



Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL DE FI DE CARRERA

TÍTOL DEL TFC: Anàlisi i proves de DASH (Dynamic Adaptive Streaming over HTTP)

TITULACIÓ: Enginyeria Tècnica de Telecomunicació, especialitat Telemàtica

AUTOR: Daniel Padilla Carrasco

DIRECTOR: David Rincón Rivera

DATA: 22 de juny de 2012

Títol: Anàlisi i proves de DASH (Dynamic Adaptive Streaming over HTTP)

Autor: Daniel Padilla Carrasco

Director: David Rincón Rivera

Data: 22 de juny de 2012

Resum

En els últims anys hem viscut un creixement exponencial d'Internet i la generalització de l'accés a la xarxa a través de terminals mòbils, que permeten a l'usuari rebre fluxos audiovisuals en streaming. En paral·lel, la millora de la tecnologia ha permès millorar els codificadors i aquests han permès un millor aprofitament de l'ample de banda. Tot i així, davant d'un escenari en què hi ha diferents amplituds de banda, diferents resolucions de pantalla i característiques del terminal, l'streaming com el coneixíem fins ara (una sola qualitat independentment de les característiques del terminal, i una taxa constant o variable, però que no s'adapta a les condicions del canal) ha quedat obsolet.

Fa un parell d'anys que s'està utilitzant el concepte de streaming adaptatiu, un concepte que implica ajustar el flux audiovisual a les necessitats de cada terminal i a les característiques del canal. Apple, Microsoft o Adobe són algunes de les empreses que han desenvolupat el seu streaming adaptatiu. Tanmateix, totes aquestes solucions són propietàries. Dynamic Adaptive Streaming over HTTP (DASH) és un nou estàndard de la família MPEG que neix amb la intenció d'estandarditzar i unificar aquestes solucions.

Aquest treball ha consistit en entendre i descriure l'estàndard DASH, fer proves amb el software disponible, i treure'n conclusions. Entre les eines disponibles hi ha segmentadors, reproductors i validadors, que hem configurat per comprovar el funcionament de DASH amb diversos escenaris de proves. Per emular el comportament hostil de la xarxa s'ha utilitzat l'emulador Dummynet.

S'ha aconseguit crear streams DASH personalitzats amb els diversos segmentadors, reproduir correctament l'stream amb els reproductors DASH, i hem verificat que només alguns d'aquests són realment capaços d'adaptar-se a les condicions de la xarxa. S'ha observat també les mancances de l'estàndard, com ara el contingut live o la reproducció d'uns casos especials de perfils.

Title: Analysis and testing of DASH (Dynamic Adaptive Streaming over HTTP)

Author: Daniel Padilla Carrasco

Director: David Rincón Rivera

Date: June, 22th 2012

Overview

We all have seen the exponential growth of the Internet and the generalization of mobile terminals for accessing the services provided by the network, such as audio and video streaming. Moreover, the technology improvements have allowed the improvement of coders, in the direction of a more efficient use of bandwidth. Therefore, the current scenario for streaming services over IP includes many screen resolutions and different terminal characteristics. This is an environment that current streaming protocols (capable of handling only one resolution, independently of the characteristics of the terminal, and a single rate, either variable or constant, but unable to adapt to the changes in the transmission channel) cannot cope with.

Two years ago the concept of adaptive streaming was launched commercially. This concept includes the adaptation of the stream to both the terminal capabilities and the channel characteristics. Apple, Microsoft and Adobe have developed proprietary implementations of the concept, thus fragmenting the market. Dynamic Adaptive Streaming over HTTP is the new MPEG standard that tries to standardize and unify these solutions.

This thesis has focused on understanding the DASH standard, set up testbed scenarios for testing the new DASH-compliant software, and extract conclusions regarding its functionality and performance, using currently available tools such as segmenters, players and validators. We have also used Dummynet for emulating the hostile behavior of the network transmission channel.

We have been able to create personalized DASH streams with different segmenters, and receive and play them correctly with the players. Some of the players are able to adapt to network conditions, while others are not. We have also had difficulties with *live* content and some specific profiles.

Dedicatòria

Vull agrair a molta gent el poder haver fet aquest projecte. Començant per la meva família que ha tingut una paciència divina esperant a que per fi acabés el treball de fi de carrera.

A David Rincón que m'ha guiat i ajudat molt. També m'ha donat una oportunitat després de fer altres intents de TFC.

Però també els hi agrairé a la gent que moltes vegades m'han sorprès amb coses que m'han ajudat en aquest projecte: el grup oficial de DASH. I és que hi ha hagut moments en que gràcies a alguna notícia o desenvolupament seu, he pogut passar un problema.

ÍNDEX

INTRODUCCIÓ	1
CAPÍTOL 1. CONCEPTES BÀSICS	3
1.1. MPEG-2	3
1.1.1. Informació general	3
1.2. Transport Stream	3
1.2.1. Informació general	3
1.2.2. Funcionament bàsic	4
1.3. MPEG-4	5
1.3.1. Informació general	5
1.4. ISO Base Media File Format	6
1.4.1. Informació general	6
1.4.2. Funcionament bàsic	6
1.5. Streaming adaptatiu	8
1.5.1. Localització temporal	8
1.5.2. Mobilitat	8
1.5.3. Les noves tecnologies	8
1.5.4. El nou estàndard	9
1.5.5. Content Delivery Networks	10
CAPÍTOL 2. DYNAMIC ADAPTIVE STREAMING OVER HTTP (DASH) ...	11
2.1. Estructura de l'estàndard	11
2.1.1. El nucli	11
2.1.2. El servidor	11
2.1.3. El protocol de transmissió	11
2.1.4. Possibilitats	12
2.2. Entrant en l'estàndard	12
2.2.1. Informació general de L'MPD	12
2.2.2. Exemple de Media Presentation Descriptor	13
2.3. Situació actual de DASH	15
2.3.1. Evolució durant el treball	15
2.3.2. Situació del software de referència	15
2.4. Relació entre MMT i DASH	16
2.4.1. Origen de DASH	16
2.4.2. Semblances i diferències	16
2.4.3. Problemes amb definicions	17
CAPÍTOL 3. EINES	18
3.1. Sistemes operatius	18
3.2. VLC - VideoLan Client	18
3.2.1. Informació general	18
3.2.2. Evolució del mòdul DASH	18

3.2.3.	Altres canvis en el mòdul DASH.....	19
3.2.4.	Activació del mòdul.....	19
3.2.5.	Carregant streams DASH.....	20
3.2.6.	Logging amb VLC.....	20
3.3.	MP4Client	21
3.3.1.	Informació general.....	21
3.3.2.	Avantatges i inconvenients.....	21
3.4.	x264 22	
3.4.1.	Informació general.....	22
3.4.2.	Utilització de l'x264.....	23
3.5.	MP4Box.....	23
3.5.1.	Informació general.....	23
3.5.2.	Utilització	23
3.6.	DASHEncoder	24
3.6.1.	Informació general.....	24
3.6.2.	Utilització	24
3.7.	DASH server.....	25
3.7.1.	Informació general.....	25
3.7.2.	Configuració.....	25
3.8.	Dummynet i FrenzyBSD.....	26
3.8.1.	Informació general.....	26
3.8.2.	Utilització	26
3.8.3.	Pipes.....	26
3.8.4.	Configuració de Pipes	27
3.8.5.	Emular el jitter.....	27
3.9.	WireShark.....	28
3.9.1.	Informació general.....	28
3.9.2.	Utilització	28
3.10.	MPEG-DASH MPD Validator	29
3.10.1.	Informació general.....	29
3.10.2.	Validació de fitxers locals.....	30
3.10.3.	Validació de fitxers en servidor.....	30
3.10.4.	Validació de text	30
3.10.5.	Canvi d'esquema de validació.....	31
3.10.6.	La validació.....	31
3.11.	LibDASH	32
3.11.1.	Informació general.....	32
3.11.2.	Utilització	33
3.11.3.	Valoració.....	33
3.12.	DASH-JS i Google Chrome.....	34
3.12.1.	Informació general.....	34
3.12.2.	Configuració de Google Chrome per DASH.....	34
3.12.3.	Utilització	34
3.12.4.	Novetats.....	35
3.13.	Mozilla Firefox.....	36
3.13.1.	Informació general.....	36
3.13.2.	Activació	36
3.13.3.	Utilització	37

CAPÍTOL 4. PROVES REALITZADES	38
4.1. Instal·lació bàsica.....	38
4.1.1. Compilacions i patches de VLC.....	38
4.1.2. VLC 2.0.....	38
4.2. Muntant l'escenari Dummynet	39
4.2.1. Informació general	39
4.2.2. Intent d'instal·lació	39
4.3. Primeres proves.....	40
4.3.1. Primers resultats.....	40
4.3.2. Caching.....	40
4.4. Proves amb servidors	41
4.4.1. Descàrrega de fitxers precodificats	41
4.4.2. Logging amb VLC.....	41
4.4.3. Desxifrant informació de logging	41
4.4.4. Altres proves.....	42
4.5. Creant els DASH stream	42
4.5.1. Preparant l'escenari.....	42
4.5.2. MP4Box	43
4.5.3. DASHEncoder	43
4.6. Utilitzant Dummynet.....	44
4.6.1. Preparant l'escenari.....	44
4.6.2. Comprovant el funcionament de DASH	45
4.6.3. Preparant un altre escenari	46
4.6.4. Noves instal·lacions.....	46
4.6.5. Proves amb l'ample de banda	46
4.6.6. Provant amb el retard	49
4.6.7. Provant amb pèrdues	51
4.6.8. Provant amb jitter	51
4.6.9. Anàlisi del diàleg HTTP	51
4.6.10. DASH-JS amb Google Chrome.....	53
4.7. Altres proves.....	54
4.7.1. Contingut "Live"	54
4.7.2. Mozilla i LibDASH.....	54
4.7.3. WebM	54
CAPÍTOL 5. CONCLUSIONS I LÍNIES FUTURES	55
5.1. Conclusions	55
5.2. Impacte Mediambiental	55
5.3. Línies futures	56
BIBLIOGRAFIA	57
GLOSSARI.....	61
ANNEXOS.....	62
A.I. MPEG MEDIA TRANSPORT (MMT)	62

A.I.1.	Informació general	62
A.I.2.	Característiques	62
A.I.3.	Evolució de l'estàndard	63
A.I.4.	Entrant en l'estàndard	64
A.II.	Media Presentation Descriptor.....	67
A.II.1.	Exemple de l'estudiant	67
A.III.	DASHEncoder	70
A.III.1.	El fitxer de configuració	70

ÍNDEX DE TAULES

TAULA 1.1. COMPARATIVA DE TECNOLOGIES DE STREAMING ADAPTATIU	9
TAULA 3.1. COMANDES DE MP4BOX.....	24
TAULA 3.2. OPCIONS UTILITZADES AMB DUMMYNET	27
TAULA 4.1 RESUM DE LA PRIMERA PROVA SOBRE VLC.....	40

ÍNDEX DE FIGURES

FIG. 1.1 EXEMPLE DE L'ÚS DE PIDS I DE LES TAULES EXTRET DE [3]	5
FIG. 1.2 ESTRUCTURA DE ISOBMFF. EXTRET DE [5]	7
FIG. 1.3 DRETA: TOPOLOGIA AMB UN SOL SERVIDOR. ESQUERRA: TOPOLOGIA AMB UNA CDN. EXTRET DE [7].....	10
FIG. 2.1 ELEMENTS DE DASH. EXTRET DE [6].....	12
FIG. 2.2 ESTRUCTURA DEL FITXER .MPD. EXTRET DE [8]	13
FIG. 2.5 MP4CLIENT	15
FIG. 2.6 VLC AMB EL MÒDUL DASH.....	16
FIG. 3.1 ACTIVACIÓ DEL MÒDUL DASH A PARTIR DE LA VERSIÓ 2.0 DE VLC	19
FIG. 3.2 REPRODUINT UN FITXER MPD.....	20
FIG. 3.3 ACTIVANT EL LOG DE VLC.....	20
FIG. 3.4 CREANT UN FITXER DE LOG.....	21
FIG. 3.5 CANVI DE RESOLUCIÓ AMB MP4CLIENT	22
FIG. 3.6 CAS SENZILL DE DUMMYNET.....	26
FIG. 3.7 CAPTURAR FLUX DE DADES AMB WIRESHARK.....	29
FIG. 3.8 FILTRAT DEL FLUX PER FILTRE	29
FIG. 3.9 VALIDACIÓ D'MPD PER FITXER.....	30
FIG. 3.10 VALIDACIÓ D'MPD PER URI	30
FIG. 3.11 VALIDACIÓ D'MPD PER TEXT	31
FIG. 3.12 OPCIÓ DE DIFERENTS ESQUEMES XSD	31
FIG. 3.13 COMPARACIÓ ENTRE VALIDACIONS D'MPD	32
FIG. 3.16 HABILITAR DASH-JS EN GOOGLE CHROME.....	34
FIG. 3.17 EXEMPLE DE REPRODUCCIÓ DASH SOBRE HTML5.....	35
FIG. 3.18 AMPLE DE BANDA	36
FIG. 3.19 ACTIVACIÓ DE DASH EN FIREFOX.....	37
FIG. 4.1 FINESTRA DE LOGGING DE VLC	42
FIG. 4.2 CONFIGURACIÓ DEL PRIMER TESTBED.....	45
FIG. 4.3 FUNCIONAMENT NORMAL DE DASH	45
FIG. 4.4 CONFIGURACIÓ DEL SEGON TESTBED	46
FIG. 4.5 AMPLE DE BANDA SELECCIONAT	47
FIG. 4.6 BAIXADA D'AMPLE DE BANDA.....	48
FIG. 4.7 INFORMACIÓ DE L'STREAM DE 10 MBPS AMB MEDIAINFO.....	48
FIG. 4.8 PUJADA D'AMPLE DE BANDA	49
FIG. 4.9 ERRORS PER DELAY EN LA TRANSMISSIÓ	50
FIG. 4.10 COMANDES GET DE HTTP.....	52
FIG. 4.11 GRÀFIC DE L'AMPLE DE BANDA.....	52
FIG. 4.12 EXEMPLE DE CAPTURA DE WIRESHARK	53
FIG. 4.13 MOZILLA I DASH EN ANDROID	54

INTRODUCCIÓ

En els darrers últims anys hem viscut un creixement exponencial d'Internet amb avenços com ara les millores en velocitat en les ofertes ADSL i la generalització de l'accés a la xarxa per terminals mòbils. En paral·lel, la millora dels codificadors per comprimir al màxim tot tipus de continguts multimèdia, ha fet que el consum de vídeo a través d'Internet hagi disparat les seves xifres de manera substancial.

Fins fa poc, l'estàndard més estès en compressió de vídeo i àudio era MPEG-2 (Moving Picture Experts Group), i el contenidor¹ més habitual el Transport Stream (TS). Tanmateix, l'aparició de noves eines, la millora continua de la tecnologia i la possibilitat de fer volums de càlcul que fins ara eren impensables, fan que dia a dia apareguin noves maneres molt més òptimes de transmetre multimèdia, com ara MPEG-4/H.264 i el seu contenidor Network Abstraction Layer (NAL).

D'altra banda, un dels camps en els que últimament hi ha més activitat és el de l'streaming adaptatiu, una tècnica que consisteix en millorar la qualitat de l'experiència (QoE) de l'usuari mitjançant l'adaptació de l'stream a cada dispositiu i/o tipus de connexió. Les solucions d'streaming adaptatiu disponibles al mercat són propietàries; és a dir, solucions tancades i patentades per una empresa. Hi ha una lluita continuada entre les empreses més importants del sector per imposar-se davant de les empreses rivals, i al final ens trobem amb multinacionals com Apple, Microsoft i Adobe, barallant-se per guanyar una lluita de patents que no fa cap bé des del punt de vista de l'usuari.

Aquest treball intentarà mostrar la capacitat de l'estàndard que ha sorgit per unificar totes les propostes que hi ha actualment sobre *streaming* adaptatiu. DASH (Dynamic Adaptive Streaming over HTTP) sorgeix de la necessitat comuna de tenir un estàndard en aquest sector.

Al llarg del TFC es descriuran els coneixements necessaris per entendre aquest estàndard i es mostraran les eines disponibles avui. Cal esmentar que aquest treball finalitza en una etapa on l'estàndard s'acaba de publicar, i les eines que hi ha en el moment de presentar aquest treball encara són escasses.

Els objectius inicials del treball també incloïen, a més de l'estàndard esmentat, l'estàndard MMT (MPEG Media Transport) com a possible objecte d'estudi, ja que tant l'un com l'altre estaven en fases molt inicials. MMT està cridat a convertir-se en el contenidor successor de TS i NAL, i també inclou funcions pròpies de protocols relacionats amb l'streaming sobre Internet, com ara RTP (Real-Time Protocol) i TCP (Transmission Control Protocol), entre d'altres característiques. Malauradament, ni l'estàndard ha finalitzat encara el seu procés d'aprovació, ni va arribar a sortir prou material d'estudi d'MMT per

¹ Un contenidor és una estructura de dades que organitza, particiona, multiplexa, sincronitza i protegeix els fluxos multimèdia, permetent-ne l'emmagatzematge en fitxer o la transmissió.

poder-lo presentar en aquest treball, però sí que s'ha pogut preparar una descripció general que es presenta a l'annex A.I.

Aquest treball està organitzat com es descriu a continuació. Es començarà amb un repàs breu de les bases necessàries per comprendre com funciona l'estàndard que volem estudiar. DASH intenta aprofitar l'experiència de diferents estàndards i la potència i l'adaptabilitat dels nous estàndards. Per entendre-ho, seran necessaris uns apunts bàsics.

Un cop acabada aquesta petita part introductòria, al segon capítol li tocarà el torn a DASH. Amb aquest capítol es pretén que el lector conegui l'estructura bàsica de DASH així com el seu funcionament general.

En el tercer capítol, més pràctic, es presenta tot un seguit d'eines que s'han utilitzat durant la creació dels diversos escenaris de proves.

El quart capítol descriu els escenaris de proves i els experiments realitzats. S'han preparat diversos escenaris tenint en compte les eines utilitzades al tercer capítol, de manera que podem trobar diversos escenaris preparats per VideoLan amb el suport de l'emulador de xarxes Dummynet, proves amb diversos navegadors i diversos resultats obtinguts.

Finalment l'últim capítol es dedicarà a les conclusions que s'han extret sobre el treball, i possibles línies per continuar allà on aquest treball no va poder arribar.

CAPÍTOL 1. CONCEPTES BÀSICS

A continuació presentem alguns conceptes bàsics previs per entendre DASH.

1.1. MPEG-2

1.1.1. Informació general

El recent creixement en la demanda d'informació per part dels usuaris sobre qualsevol tipus de tecnologia ha deixat totalment enrere patrons tecnològics que van ser creats al seu dia per una única funció específica. De fet, actualment, el consum de multimèdia per part de la població no hagués estat possible sense estàndards que han propiciat una millora substancial en diferents aspectes.

El comitè MPEG [1] s'ha encarregat, juntament amb altres grups, d'estandarditzar una sèrie d'operacions que faciliten i simplifiquen a l'usuari el consum de multimèdia. Gràcies a aquests grups, avui dia podem gaudir d'una melodia en *streaming*, podem veure i descarregar un vídeo en alta definició o fins i tot podem substituir un canal analògic per uns quants digitals. Tot això és gràcies als estàndards, o parts d'ells, com l'MP3, MPEG-4 o MPEG-2 Transport Stream.

Molts dels estàndards actuals s'han aprofitat de l'èxit i la bona acceptació del MPEG-2 Transport Stream (TS). Aquest estàndard, l'MPEG-2, té fins 11 apartats, explicant en cada un d'ells diverses tecnologies. El Program Stream (PS) i el TS es descriuen com les dues solucions en l'àmbit de sistema. La primera solució descriu una sèrie de normes per guardar els fitxers multimèdia al disc dur, mentre que la segona descriu normes per enviar els fluxos multimèdia a través de canals de comunicació (radiodifusió DVB, xarxa IP) en forma de paquets. Aquesta part va ser completament acurada i acceptada en un moment (inici de la dècada dels 1990) on Internet començava a formar part de les nostres vides.

DVB, la organització d'estandardització per la televisió digital [2], va incloure el TS com a part fonamental del seu estàndard per les xarxes terrestre, cable i satèl·lit. El TS segueix creixent, i fins a dia d'avui, encara és altament utilitzat. Però les tecnologies avancen, i el TS s'ha quedat curt en molts aspectes.

1.2. Transport Stream

1.2.1. Informació general

MPEG-2 va revolucionar l'indústria de les telecomunicacions amb la definició del contenidor Transport Stream (TS), que va cobrir una àrea que va resultar ser tot un pas endavant en les comunicacions.

A diferència del Program Stream (PS), que es basa en una estructura de fitxer (amb un inici i un final, les capçaleres al principi de les dades, i poca protecció contra errors), el Transport Stream protegeix molt la informació i la parteix en unitats molt petites amb capçaleres independents, el que permet ser transmeses en flux o stream.

El TS s'adapta més a les xarxes de transmissió (radiodifusió terrestre, satèl·lit, cable o xarxa IP), que són un medi molt més hostil (en quant a pèrdues, retard i variació del retard o *jitter*) que no pas el disc dur d'un ordinador.

1.2.2. Funcionament bàsic

El TS no només va introduir una paquetització de la informació si no que a més, va introduir un concepte més avançat, definint una sèrie de estructures de dades especials anomenats taules, que permeten senyalitzar el contingut dels fluxos multiplexats a un TS. D'aquesta manera, dins d'un mateix TS es poden trobar diferents fluxos de dades que avancen en paral·lel, cadascun a una taxa diferent, durant la transmissió. Aquest concepte s'ha utilitzat àmpliament en televisió, on un Transport Stream equival a un canal de DVB i on cada servei és definit mitjançant diverses taules, que a la seva vegada apunta a cadascun dels fluxos elementals (vídeo, àudio, teletext) que es multiplexen al flux TS.

Al TS pot haver diverses taules específiques com la PAT (Program Association Table) o la PMT (Program Map Table), creades per la descripció d'un servei concret, entenent servei com a conjunt de flux de vídeo, àudio i dades que formen un únic programa. Totes aquestes taules van identificades per un únic PID (Program ID) específic. Algunes taules generals com ara la PAT tenen un PID determinat (0x00 en el cas de la PAT), mentre que els altres fluxos poden tenir qualsevol número de PID (dintre del rang 0-8191, que correspon als 13 bits del camp PID). El TS s'estructura a partir de la taula amb PID 0x00 (PAT), i és en aquesta taula bàsica on es recullen les PMT que estructuraran els diferents serveis indicant a cadascuna els fluxos de dades amb diferents PIDs com es pot observar a la figura 1.1.

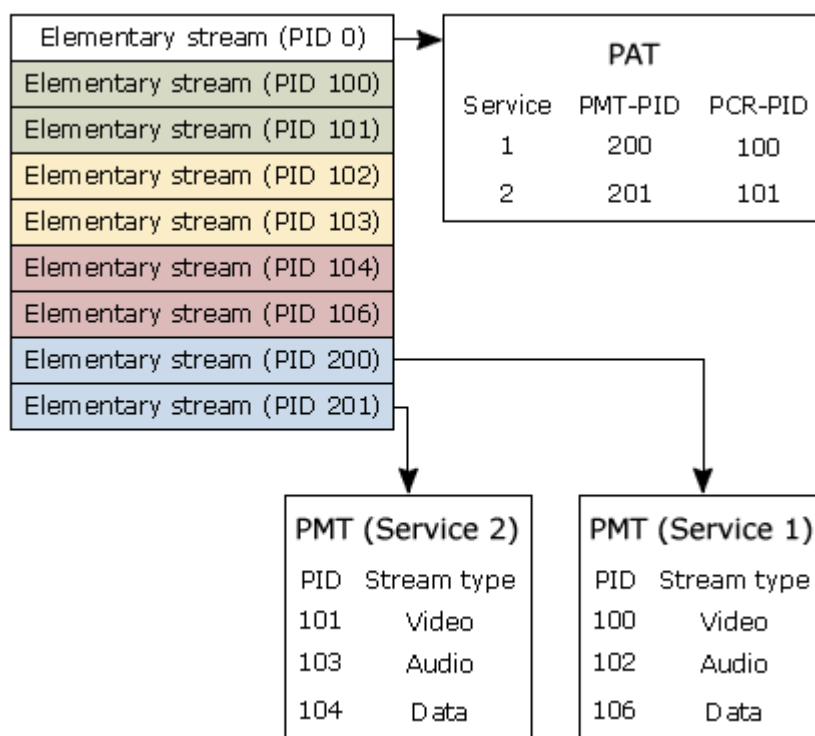


Fig. 1.1 Exemple de l'ús de PIDs i de les Taules Extret de [3]

1.3. MPEG-4

1.3.1. Informació general

MPEG-2 ha evolucionat cap a l'MPEG-4. El canvi ha estat molt gran, i encara que s'ha seguit utilitzant moltes de les idees i paradigmes introduït per un èxitós MPEG-2, s'han introduït idees noves i sobre tot, en un àmbit de creixement tecnològic, una actualització en quant a l'aprofitament d'una capacitat computacional més gran.

És que el problema més important en quant a un estàndard de codificació i descodificació de vídeo com és l'MPEG, és la limitació de la capacitat de càlcul. A mesura que avança la tecnologia, un ordinador es capaç de fer moltes més operacions per segon i per tant, és capaç de fer operacions molt més complexes. Això implica que el que abans no es podia fer amb MPEG-2 perquè seria impossible reproduir un vídeo, actualment és factible amb les CPUs disponibles avui dia.

El problema de l'obsolescència d'un estàndard és la capacitat computacional que requereix. De fet, actualment es podria fer molt més per comprimir un vídeo o àudio, matemàticament, del que s'està fent actualment en el mercat, ja que fer un codificador amb segons quines operacions seria massa car, i això és el que ha passat amb l'estàndard MPEG-2. Operacions que abans no eren factibles, amb l'evolució de la tecnologia es tornen més assequibles gràcies a la millora de la capacitat computacional i s'estandarditza l'MPEG-4 aprofitant aquesta nova capacitat.

A més, l'estàndard MPEG-4 introdueix un nou format per guardar els fitxers digitals codificats i pretén que sigui un fitxer vàlid, no només per MPEG-4 sinó per un àmbit general. A continuació donem alguns detalls sobre un nou contenidor definit per MPEG-4.

1.4. ISO Base Media File Format

1.4.1. Informació general

ISO Base Media File Format (ISOBMFF) [4] és el format introduït a MPEG-4, encara que també apareix en JPEG2000 (Joint Photographic Experts Group). Un fitxer amb format ISOBMFF està completament estructurat per caixes², de manera que una caixa pot contenir altres caixes a l'interior seu, apuntar a altres caixes (en un altre fitxer per exemple) o simplement contenir paquets de dades anomenats chunks.

1.4.2. Funcionament bàsic

En ISOBMFF existeix una caixa base a partir de la qual s'anirà organitzant l'estructura del fitxer. En aquest cas la caixa *ftyp* és la base que defineix com està estructurat el fitxer. A partir d'aquí ens trobem la caixa *moov* que conté la informació referida al vídeo i *mdat* que conté les dades. Segons de quin contenidor estiguem parlant i com estigui estructurat aquest vídeo trobarem unes caixes o unes altres.

La figura 1.2 il·lustra el contingut estructurat de caixes que té un MP4. Comencem pel box *ftyp* on trobarem les dades més bàsiques del fitxer, que identifiquen el tipus de fitxer i els formats compatibles. Amb aquesta petita caixa, en identificar el tipus de fitxer, el descodificador ja pot saber quin tipus de caixa es pot esperar i pot tenir una idea aproximada de com estan estructurades les caixes. A partir d'aquí el descodificador trobarà la caixa *moov*, on es descriuen on trobar les caixes de dades i informació vària del vídeo, i dins d'aquesta:

- La caixa *mvhd* defineix la capçalera del vídeo amb informació varia com la durada, resolució, entre d'altres.
- Les caixes *trak* defineixen un flux de dades determinat (àudio, vídeo, etc) marcant la localització en l'arxiu i la durada del flux. Aquesta caixa engloba a altres com:
 - Exactament una *tkhd* que conté informació de les característiques de la pista
 - Pot contenir una *edts* per redefinir la línia de temps d'una pista.

² Un box o caixa en ISOBMFF és un conjunt de dades de longitud determinada pels 4 primers bytes per contenir un tipus específic de dades.

- I finalment ha de contenir una *mdia* box per contenir altres objectes important.
 - La caixa *minf* conté altres caixes d'informació de la pista.
 - Una *vmhd* informant del color i mode dels gràfics
 - Una caixa *dinf* que declara la localització de les dades de la pista
 - Finalment una caixa *stbl* que crea una taula amb la grandària de les dades de manera que es puguin separar les dades.
- La caixa *moov* també pot contenir la caixa *udta* amb informació de l'usuari en les caixes:
 - Pot contenir diverses caixes opcionals com *meta*, *covr* o *©nam* amb diferent tipus de dades (XML, imatge de la cobertura de la pista etc).

Finalment trobarem la caixa *mdat* on s'engloben les dades de tot l'arxiu multimèdia.

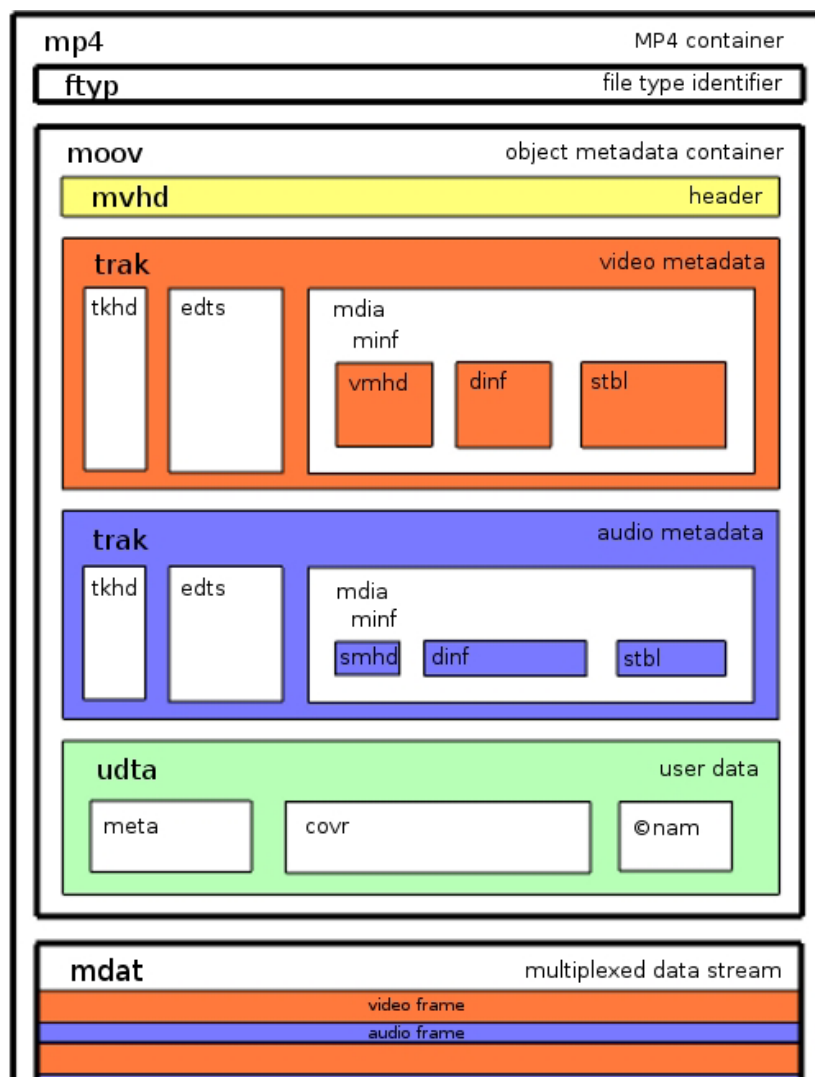


Fig. 1.2 Estructura de ISO BMFF. Extret de [5]

No tots els fitxers tenen aquesta informació, i moltes de les taules aquí explicades poden no aparèixer en un fitxer mp4.

1.5. Streaming adaptatiu

1.5.1. Localització temporal

El gran creixement que han sofert els smartphones en els darrers anys ha estat impulsat en gran mesura per la introducció de potents sistemes operatius com ara Android o IOS, capaços d'oferir a l'usuari una diversitat d'aplicacions a la mateixa altura que un ordinador, amb una potencia res envejable a l'ordinador i amb una mobilitat molt superior a aquest. Però no només en l'àmbit de tecnologies mòbils es centra el nou estàndard: el fet que hi hagin velocitats de connexió a la xarxa molt diferents entre els usuaris, que la topologia de la xarxa sigui una incògnita i per tant una connexió fixa pugui tenir fluctuacions en l'ample de banda, fa que el nou estàndard pugui ser utilitzat per un grup molt ampli d'usuaris. En definitiva, sorgeix una nova idea de transmetre streaming, es crea una nova tecnologia per que totes aquestes diferències entre terminals siguin mínimes, oferint en qualsevol cas un flux multimèdia.

Encara que l'streaming adaptatiu potencia la mobilitat dels terminals, també es desenvolupa amb la intenció de reaprofitar recursos com proxies, Content Delivery Networks (CDN), tallafocs, entre d'altres que estan molt estesos en la xarxa actual.

1.5.2. Mobilitat

El fet que els smartphone tinguin una mobilitat molt superior, ha fet que la demanda de multimèdia hagi crescut en sobre mesura en els darrers anys. La possibilitat de poder arribar a veure vídeos allà on es vulgui ha suposat un gran impacte en l'ús d'internet en l'àmbit mundial. Esperant a l'autobús, d'acampada a la muntanya, prenent una copa al bar, en qualsevol lloc tenim disponible una connexió més que suficient per gaudir de tot tipus de multimèdia. Però la mobilitat introdueix el factor d'una cobertura variable i d'unes condicions canviant del canal de transmissió, així que, la tecnologia ha volgut avançar per donar a aquests usuaris una qualitat de servei que fins ara no tenien.

1.5.3. Les noves tecnologies

Han aparegut diverses tecnologies que miren de solventar els problemes esmentats. Microsoft Smooth Streaming (MS-SSTR) surt com a proposta de Microsoft per les olimpíades de l'estiu 2008. Apple HTTP Live Streaming (HLS) va ser llançat per donar un streaming de qualitat al producte estrella d'Apple, l'iPhone 3G, durant l'abril de 2008. Finalment, Adobe HTTP Dynamic Streaming (HDS) surt per l'any 2010. Però totes aquestes tecnologies lluitaven entre elles per aconseguir més popularitat, així que MPEG va decidir estandarditzar una

nova tecnologia amb idees comunes amb Dynamic Adaptive Streaming over HTTP (DASH). Aquestes tres noves formes de reproduir fluxos multimèdia han estat les més populars en els últims 2 anys. En la taula 1.1 es pretén fer un resum de les característiques més importants d'aquestes tecnologies.

Taula 1.1. Comparativa de tecnologies de streaming adaptatiu

Opció	MS-SSTR	HLS	HDS	DASH
Protocol	HTTP	HTTP	RTMP	HTTP
Contenidor	MP4	MPEG-2 TS	MP4,FLV	MP4
Tipus d'arxiu en servidor	Un sol fitxer (per bitrate)	Fitxers fragmentats	Un sol fitxer (per bitrate)	Un sol fitxer (per bitrate) o fitxers fragmentats
Format del manifest³	ism (XML) i ismc (XML)	m3u8 (Propietari)	F4M (XML)	MPD (XML)
Servidor	IIS Media Server o mòdul extra per Apache	Qualsevol servidor HTTP	Flash Media Server o mòdul extra per Apache	Qualsevol servidor HTTP
Llicència	Propietària	Internet draft	Propietària	Estàndard ISO/IEC 23009-1

1.5.4. El nou estàndard

DASH, definit a l'estàndard ISO/IEC 23009-1 [6] neix per mirar de solventar les diferents condicions que es pot trobar un terminal i per unificar totes les idees que apareixen en les tecnologies mencionades.

DASH es va idear amb la intenció d'oferir un servei d'streaming adaptatiu sobre el protocol HTTP. Aquest detall és important, ja que la majoria de serveis i aplicacions actuals estan pensats principalment per HTTP. Proxies, CDN i tallafocs estan àmpliament utilitzats per aquest protocol fent que moltes eines que fins ara ofereixen un inconvenient ara siguin un avantatge. Per exemple, no s'ha de configurar de nou un tallafocs, ja que el tràfic és purament HTTP. El mateix passaria amb altres eines: els Network Address Translators (NAT) normalment tenen els ports d'entrada i sortida per HTTP i els proxies continuarien oferint aquest servei per DASH.

³ Un manifest és l' arxiu on s'especifica les característiques d'un objecte. El concepte de manifest és molt utilitzat en l'actualitat en diferents sectors tecnològics. En el cas de l'streaming, el manifest determinarà la durada de la font, els fitxers físics multimèdia i altres dades.

1.5.5. Content Delivery Networks

Un cas d'especial interès és els de les CDN. Aquesta tecnologia que permet reduir latència i maximitzar l'ample de banda de descàrrega dels continguts, entre d'altres coses, es basa en la idea de tenir diversos servidors distribuïts per la xarxa en comptes d'un.

Fins fa relativament poc, la idea més comuna de servidor era la d'un ordinador central al qui qualsevol persona podia accedir. Això, per començar, crea un coll d'ampolla important en aquest servidor. També, al ser un servidor únic, la latència pot ser excessiva depenent del servei i el lloc físic del servidor respecte a l'usuari. Finalment, l'ample de banda de pujada del servidor és compartit per tots els usuaris.

La CDN millora aquestes característiques mitjançant una distribució de servidors per tota la xarxa. Cada servidor substitut de la CDN actua com una mena de proxy, demanant al servidor original les dades i copiant-les de manera que el seu contingut pugui ser accedit, més tard, des del servidor substitut. Aquesta topologia, al estar distribuïda per tot el món, permet millorar la latència, i al haver diferents servidors, els colls d'ampolla i l'ample de banda de pujada milloren.

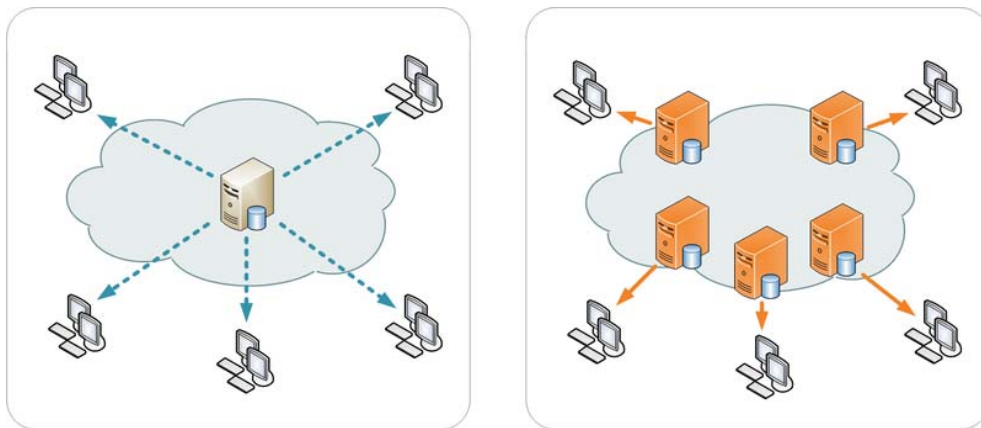


Fig. 1.3 Dreta: Topologia amb un sol servidor. Esquerra: Topologia amb una CDN. Extret de [7]

Les CDN admeten tràfic HTTP, per tant, també en streaming adaptatiu es pot fer ús d'aquesta topologia.

CAPÍTOL 2. DYNAMIC ADAPTIVE STREAMING OVER HTTP (DASH)

Aquest capítol descriu les característiques més importants de l'estàndard DASH.

2.1. Estructura de l'estàndard

2.1.1. El nucli

Totes les dades que DASH necessita per oferir adaptivitat estan contingudes a un sol fitxer XML anomenat MPD (*Media Presentation Descriptor*). Aquest fitxer és qui defineix tots els paràmetres vàlids per fer streaming d'un flux determinat. Així doncs, en aquest fitxer XML tenim una llista de streams amb els seus paràmetres associats visibles. D'aquesta manera, un usuari mòbil que gaudeixi de poc ample de banda per mala cobertura pot voler escollir un ample de banda pel vídeo més baix que un usuari amb una connexió fixa que usualment sol tenir una connexió més estable amb el temps. Paràmetres com resolució de la pantalla, la mida de la cua de *buffer* (que a la seva vegada determina el retard amb què es visualitza el vídeo) o l'idioma, per exemple, també són paràmetres que el client HTTP podria arribar a escollir sempre i quan estiguin presents a l'XML.

2.1.2. El servidor

L'estructura per muntar un servidor per distribuir streaming en DASH és realment simple. Es tracta només de posar en marxa un servidor HTTP com els que estan utilitzats servint els fitxers MPD disponibles. El client descarrega el fitxer amb extensió *.mpd* i a partir d'aquí té la suficient informació sobre diferents resolucions, qualitats, amplex de banda, buffers, idiomes, etc. Per tant, un cop descarregat aquest XML el client ja sap a quina adreça ha d'anar a buscar l'stream que busca.

2.1.3. El protocol de transmissió

Al parlar d'streaming sobre HTTP en el que els enllaços estan referenciats per URLs, estem creant una estructura de fitxers pregenerats i col·locats de manera determinada al servidor HTTP. I encara que parlem de contingut dinàmic com per exemple el contingut d'una càmera web, el contingut ha de poder ser adreçat mitjançant HTTP. En el cas de la càmera web, el servidor crea un MPD dinàmic mitjançant l'ús de sessions de manera que a l'hora de rebre l'stream té una URL determinada, tanmateix, aquesta és una URL fantasma que es redirigeix cap a l'stream DASH, finalment el client només ha de descarregar-se aquest *arxiu* mitjançant el protocol HTTP. Com ja s'ha

comentat, es tracta d'una situació típica en internet on el protocol de transmissió és el ben estudiat HTTP i totes les eines (proxies, CDN, tallafocs, etc) que utilitzin aquest protocol seran vàlides per aquesta estructura, fent que el traspàs a aquesta tecnologia sigui amb un cost molt baix i podent aprofitar la gran majoria d'eines que es disposen per aquest protocol de transmissió.

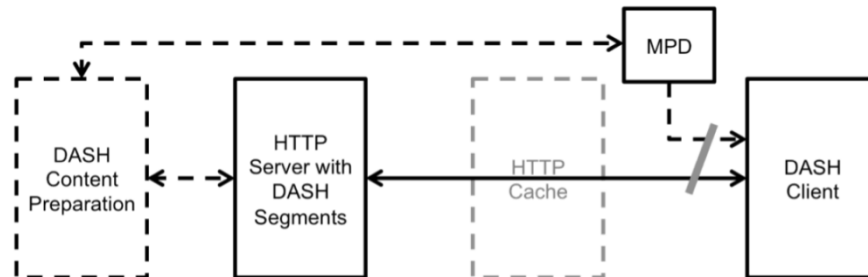


Fig. 2.1 Elements de DASH. Extret de [6]

2.1.4. Possibilitats

Aquesta nova tecnologia no només ofereix una configuració de video inicial sinó que, a més, dins d'un mateix *stream* hi ha punts d'anclatge de manera que l'usuari pot, en tot moment canviar d'un *stream* a un altre. Això possibilita canvis de velocitat sofertes per un canvi de les condicions de la xarxa IP (per canvis en cobertura del mòbil o variació de la congestió a la xarxa fixa, per exemple), l'adaptació de qualitats per absorbir l'inici d'un stream començant amb un segment d'ample de banda mínim, i per tant una qualitat inferior, per anar pujant a mesura que tenim prou buffer, o fins i tot un canvi de resolució al connectar un dispositiu de visualització amb una resolució major.

Com es pot veure les oportunitats que permet DASH a l'usuari estan enfocades en la nova forma d'accedir a la xarxa. Les connexions ara són més dinàmiques i és imperatiu que els estàndards s'adaptin a aquest nou escenari.

2.2. Entrant en l'estàndard

2.2.1. Informació general de L'MPD

Aquest fitxer MPD està estructurat de manera que la interpretació de l'XML és bastant ràpida. Primer de tot ens trobem amb un primer bloc on es presenta la informació general de l'stream, com ara si es tracta d'un stream en directe, "live", o sota demanda, "on demand", i si és el cas, la duració real de tot l'stream, el buffer mínim necessari per qualsevol stream, o la URL base dels streams, entre d'altres informacions.

Després d'aquest primer bloc d'informació general entrem en l'stream, organitzat en una estructura de fins a quatre capes.

- A la primera capa trobem una descripció dels períodes. El període és... El protocol ens permet definir diferents períodes per l'stream de manera que es pugui accedir o saltar d'un temps a un altre ràpidament en diferents stream.
- Dins de cada període està definit cada part de l'stream com un "adaptation set". Aquí entrem en el contingut separat en diferents medis. Podem trobar diferents àudios per a cada idioma, diferents possibilitats de còdecs, aplicacions, subtítols, etc.
- I sobre cada adaptation set es poden trobar diferents representations. Aquestes representations no són més que les diferents configuracions per cada medi. És a dir, les diferents opcions de configuració que el client tindrà a l'hora d'escollir el medi. Actualment solen estar classificats per amplitud de banda i resolucions en el cas de vídeos però segons l'esquema XSD oficial poden tenir altres classificacions (perfils, freqüència d'imatge, etc).
- Finalment, cada representation conté informació dels segments que han estat generats prèviament a l'hora de crear els fitxers pel servidor HTTP. En aquest apartat es pot trobar l'enllaç a aquests segments. Aquesta segmentació és per marcar un bloc mínim, normalment de buffer, on aquesta configuració no canviarà.

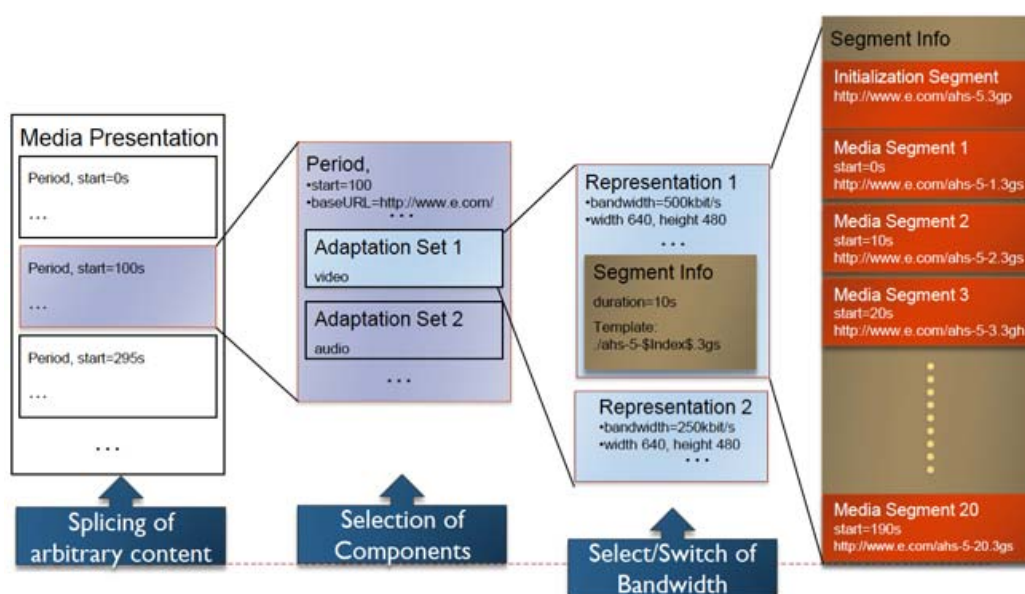


Fig. 2.2 Estructura del fitxer .mpd. Extret de [8]

2.2.2. Exemple de Media Presentation Descriptor

Encara que hi ha diversos MPDs a la web oficial, aquí s'exposarà el que probablement s'hagi utilitzat més en tot el treball: l'MPD de la pel·lícula Big Buck Bunny [9,10]. Tot i així l'extensió del fitxer és massa gran, per tant s'escurçaran aquelles parts repetitives.

```

<?xml version="1.0" encoding="UTF-8"?>
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:DASH:schema:MPD:2011"
  xsi:schemaLocation="urn:mpeg:DASH:schema:MPD:2011"
  profiles="urn:mpeg:dash:profile:isoff-main:2011"
  type="static"
  mediaPresentationDuration="PT0H9M56.46S"
  minBufferTime="PT1.0S">
  <BaseURL>http://www-itec.uni-klu.ac.at/ftp/datasets/mmsys12/BigBuckBunny/bunny_1s/</BaseURL>
  <Period start="PT0S">
    <AdaptationSet bitstreamSwitching="true">
      <Representation id="0" codecs="avc1" mimeType="video/mp4" width="320" height="240" startWithSAP="1"
        bandwidth="46986">
        <SegmentBase>
          <Initialization sourceURL="bunny_1s_50kbit/bunny_50kbit_dash.mp4"/>
        </SegmentBase>
        <SegmentList duration="1">
          <SegmentURL media="bunny_1s_50kbit/bunny_1s1.m4s"/>
          <SegmentURL media="bunny_1s_50kbit/bunny_1s2.m4s"/>
          <SegmentURL media="bunny_1s_50kbit/bunny_1s3.m4s"/>
          .
          .
          .
          <SegmentURL media="bunny_1s_50kbit/bunny_1s595.m4s"/>
          <SegmentURL media="bunny_1s_50kbit/bunny_1s596.m4s"/>
          <SegmentURL media="bunny_1s_50kbit/bunny_1s597.m4s"/>
        </SegmentList>
      </Representation>
      <Representation id="1" codecs="avc1" mimeType="video/mp4" width="320" height="240" startWithSAP="1"
        bandwidth="91932">
        <SegmentBase>
          <Initialization sourceURL="bunny_1s_100kbit/bunny_100kbit_dash.mp4"/>
        </SegmentBase>
        <SegmentList duration="1">
          <SegmentURL media="bunny_1s_100kbit/bunny_1s1.m4s"/>

```

Fig. 2.3. Exemple d'MPD. Extret de [9]

En aquest exemple, es pot veure com la primera part és el propi descriptor de l'MPD, definint paràmetres globals com l'XSD utilitzat, o el profile, entre d'altres. Després trobem informació general de l'stream. També trobem abans de començar amb els períodes la direcció del servidor i localització dels streams (BaseURL). Tal i com es mostra en la figura 2.3, la duració del segment és d'1 segon i disposem de les taxes, en Kbps (Kbits/s): 50, 100, 150, 200, 250, 300, 400, 500, 600, 700, 900, 1200, 1500, 2000, 2500, 3000, 4000, 5000, 6000 i 8000. Aquestes taxes estan repartides amb resolucions de: 320x240, 480x360, 1280x720 i 1920x1080.

Després d'aquesta informació ja trobem un període amb un AdaptationSet i diverses representacions tal i com s'ha mostrat en l'apartat 2.2.1.

Hi ha dos maneres diferents d'organitzar l'stream DASH. La primera és per fragments diferents com el mostrat a la figura 2.3, i la segona és mostrar un únic fitxer indexat i que s'especifiqui com a direcció (SegmentURL) de l'stream els segments dins d'aquest únic fitxer, com es mostra a la figura 2.4. A l'exemple el primer segment inclou des del byte 863 fins el 7113, el segon des del 7114 al 14104, i així successivament.

```
<SegmentBase>
  <Initialization sourceURL="bunny_1s_50kbit/bunny_50kbit_dashNonSeg.mp4" range="0-862" />
</SegmentBase>
<SegmentList duration="1">
  <SegmentURL media="bunny_1s_50kbit/bunny_50kbit_dashNonSeg.mp4" mediaRange="863-7113" />
  <SegmentURL media="bunny_1s_50kbit/bunny_50kbit_dashNonSeg.mp4" mediaRange="7114-14104" />
</SegmentList>
```

Fig. 2.4 MPD de fitxer no segmentat. Extret de [10]

2.3. Situació actual de DASH

2.3.1. Evolució durant el treball

Malgrat que el desenvolupament d'eines per provar la tecnologia DASH ha estat bastant escassa i caòtica durant els primers mesos de la duració d'aquest projecte, a mesura que ha passat el temps han aparegut diferents eines per poder fer més proves sobre la tecnologia DASH. Tanmateix, les dues eines principals per testejar la tecnologia han estat el software oficial de DASH [11], que no és més que un nou mòdul per al reproductor VideoLan [12] i el nou GPAC [13]. Aquestes dues aplicacions han estat el pilar bàsic durant la major part d'aquest treball tot i que han aparegut noves eines, com DASH-JS, que també s'han provat i avaluat.

2.3.2. Situació del software de referència

Actualment GPAC i el grup oficial de DASH treballen conjuntament, com a mínim parcialment. I és que el software de GPAC es va avançar en quant a la generació dels fitxers DASH pels servidors HTTP i des de llavors, el software oficial utilitza les llibreries de GPAC per la generació dels fitxers DASH. Tot i això, el software oficial deixa una mica més de "llibertat" al possibilitar una generació múltiple de diferents stream, tal com es veurà més endavant.

El software GPAC inclou per una banda el MP4Client, un software que permet la reproducció d'streams DASH, i el MP4Box, utilitzat per la generació dels streams DASH. Tanmateix, el MP4Box funciona a partir d'arxius ja codificats, és a dir, utilitzant fitxers d'extensió x264 pot crear els diferents fragments i el fitxer MPD.

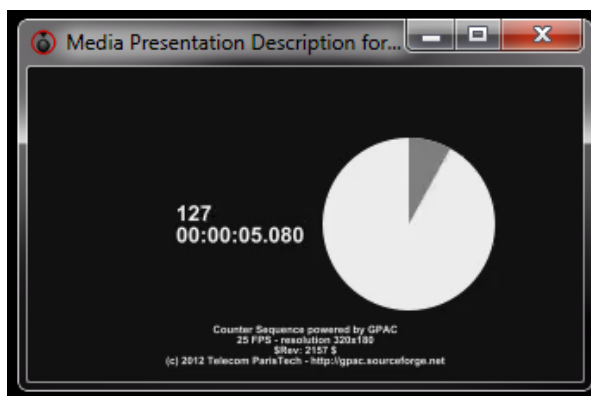


Fig. 2.5 MP4Client

Pel que fa a l'escenari al voltant de VideoLAN, es disposa d'un mòdul de visualització d'streams DASH i un executable creat pel grup oficial de DASH per la generació d'streams, DASHEncoder [14], que utilitza la llibreria x264 i el MP4Box per la generació d'streams DASH. Això dona molt més joc per poder configurar un stream correctament, ja que amb el DASHEncoder podem extreure tots els streams que vulguem a partir d'un video *raw*⁴, o qualsevol format que accepti la llibreria de codificació (en aquest cas x264).



Fig. 2.6 VLC amb el mòdul DASH

2.4. Relació entre MMT i DASH

2.4.1. Origen de DASH

Inicialment, DASH va començar com una part de MMT (*MPEG Media Transport*). De fet, DASH va ser un mòdul de MMT per donar solució a certs escenaris, aquells en què l'usuari pretén gaudir d'un servei multimèdia mitjançant la web.

2.4.2. Semblances i diferències

Tots dos tenen com a objectiu adaptar en tot moment una connexió per tal de possibilitar una bona qualitat de servei.

Però l'àmbit d'actuació és clarament el que els distància. Per una part, és clar que MMT pretén crear un tipus de tecnologia que permeti un contingut multimèdia molt més global, sense importar el tipus de xarxes, protocols o tecnologies utilitzats per les capes inferiors. DASH, en canvi, només pretén crear un streaming adaptatiu en HTTP. Es subratlla HTTP perquè des del moment en el que es pretén disposar aquesta tecnologia només pel protocol HTTP, s'està distanciant dels objectius de MMT que pretenen estar deslligats del màxim de tecnologies possible. Es poden trobar més detalls sobre MMT a l'annex A.I.

⁴ S'anomena fitxer RAW a aquell fitxer multimèdia sense cap tipus de codificació. És aquest arxiu el que sempre conté tota la informació possible d'un vídeo i per tant la qualitat més alta a expenses d'un major volum de dades.

2.4.3. Problemes amb definicions

DASH té com a origen un apartat de MMT, i aquest últim pretén ser un estàndard global, deslligat de tota tecnologia externa de manera que es pugui adaptar al futur. El fet que un MPD defineixi l'enllaç als fragments de l'stream DASH mitjançant el protocol HTTP crea, en el grup de discussió [15] sobre aquests dos estàndards, una confusió.

Durant els mesos de gener i febrer de l'any 2012, es va estar discutint sobre aquest fet, amb partidaris d'ampliar l'estàndard DASH per acceptar altres tipus de protocols de transport. També es discuteix sobre el fet que DASH no s'adequa al format d'un segment MMT, com per exemple l'encapsulament. D'altra banda estan els partidaris que DASH no es defineix exclusivament sobre HTTP, sinó que únicament defineix l'ús de URLs com a medi per enllaçar el contingut i per tant pot ser compatible amb MMT.

CAPÍTOL 3. EINES

En aquest capítol es farà una breu introducció als escenaris de proves mitjançant una llista de les aplicacions considerades importants i utilitzades durant aquest treball de fi de carrera.

3.1. Sistemes operatius

Les eines emprades per la banda del client en aquest treball s'han utilitzat sobre els sistemes operatius Windows 7 i Ubuntu 12.04 (Precise Pangolin, amb kernel 3.0.0.17 generic). Això ha estat degut a la inexistència d'alguns d'aquestes eines en algun dels sistemes operatius. Per exemple, en el cas de VLC i el mòdul de DASH, la primera compilació va ser sota Ubuntu simplement perquè no hi havia res encara per Windows. Però la llibreria libDASH [16], en canvi, que treballa sota .NET framework, està compilada per Visual Studio sota Windows.

Per l'ordinador que fa de servidor HTTP, no importa massa ja que el que necessitem és un servidor HTTP sense cap modificació addicional. Durant el treball s'han configurat dos servidors HTTP: Apache [17] i tthttpd [18]. Per motius pràctics, s'ha escollit la distribució FrenzyBSD i tthttpd, ja que aquesta distribució també conté el Dummynet [19], l'eina utilitzada a les proves de DASH descrites al capítol 4, però també s'han realitzat moltes proves amb un servidor sota Ubuntu amb Apache i un servidor extra pel Dummynet.

3.2. VLC - VideoLan Client

3.2.1. Informació general

Aquesta aplicació és el principal software utilitzat per aquest projecte. El que ens permet, bàsicament, és la reproducció d'arxius multimèdia. Sota llicència GPL, VLC [12] ha tingut un gran èxit i se'l considera un dels millors reproductors multimèdia en la actualitat.

Però el que és important per nosaltres és que VideoLan va ser escollit per ser el software de proves de la implementació oficial de DASH. El grup oficial de DASH va crear un mòdul per VLC de manera que es pogués veure streams DASH des de VLC.

Aquest mòdul ja existia quan es va començar aquest treball de fi de carrera, però ha anat evolucionant en paral·lel i canviant les seves capacitats, així com la facilitat d'instal·lació i d'ús.

3.2.2. Evolució del mòdul DASH

Mentre que al principi VLC oferia un mòdul que cada usuari havia d'incloure mitjançant la compilació de VLC introduint explícitament el mòdul, des de la

sortida de la versió 2.0 de VLC a mitjans de febrer de 2012, DASH incorpora per defecte en la instal·lació. Encara que en el seu moment es va utilitzar aquesta versió, actualment s'utilitza una *nightly build*⁵ de la versió 2.1, ja que a l'etapa final del treball s'han fet varies proves amb la versió 2.0 intentant reproduir streams DASH i no ha carregat l'stream correctament.

3.2.3. Altres canvis en el mòdul DASH

Tanmateix, no és l'única cosa que ha canviat. Si amb la versió 2.0 hem vist un endarreriment, hem trobat vídeos [20] demostrant certes capacitats d'un mòdul de DASH que mai s'han vist en aquesta versió ni en la més actual. Aspectes com ara la visualització del *Adapation Set* utilitzat segons la característica amb un bon gràfic a nivell de consola, transferència de sessions entre dispositius i opcions de configuració més avançades han desaparegut. Probablement s'ha fet això per facilitar la reproducció de streams DASH per part d'un usuari inexpert.

3.2.4. Activació del mòdul

Tot i que el mòdul DASH està inclòs en el VLC, per defecte està desactivat, així que s'ha d'activar. La manera de fer-ho consisteix en anar a **Eines>Preferències** (Mostrant la vista avançada amb totes les preferències)>**Entrada / Còdecs>Filtres de transmissió** y activar **Dynamic Adaptive Streaming over HTTP**. Després de fer aquesta operació es reinicia el VLC per assegurar la correcta implementació de la llibreria.

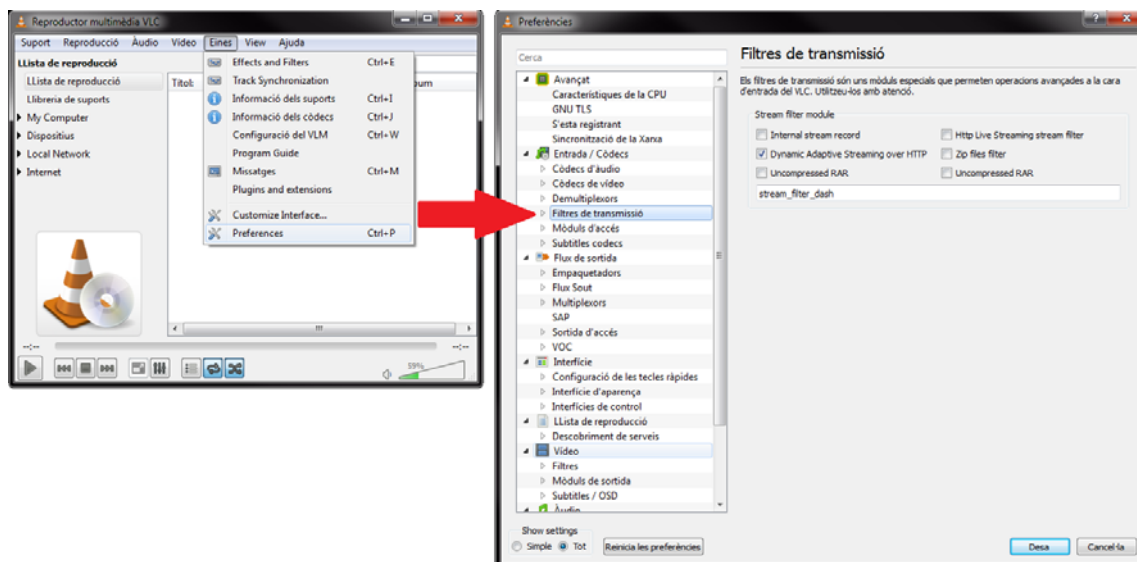


Fig. 3.1 Activació del mòdul DASH a partir de la versió 2.0 de VLC.

⁵ Una nightly build és una compilació diària d'un projecte. De manera que aquesta nightly build, encara que té novetats respecte a la versió oficial d'un software, també pot ser inestable i sensible a errors.

3.2.5. Carregant streams DASH

Per reproduir els streams DASH hem de fer: **Suport>Obre un flux de la xarxa** i aquí enganxem l'adreça d'un fitxer MPD.

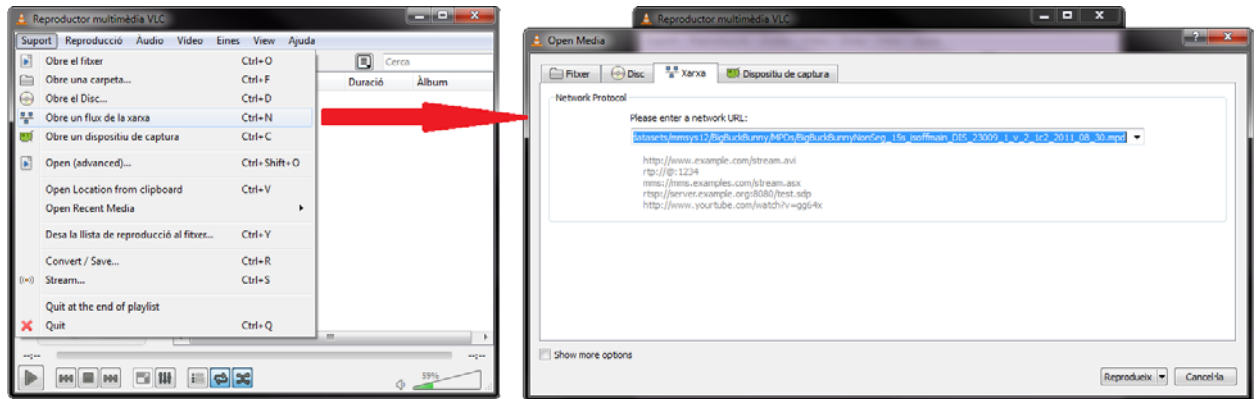


Fig. 3.2 Reproduint un fitxer MPD

3.2.6. Logging amb VLC

Però encara hi ha una opció molt interessant. Encara que no s'ha pogut trobar el gràfic a nivell de consola que permetia observar directament l'ample de banda seleccionat en tot moment, si que es pot veure en tot moment l'activitat del mòdul DASH, i amb l'eina de logging de VLC podem veure els intents del mòdul en adaptar-se el millor possible a l'ample de banda que tenim. Per activar aquesta opció, cal anar a **View>Afegir una interfície>Debug Logging**.

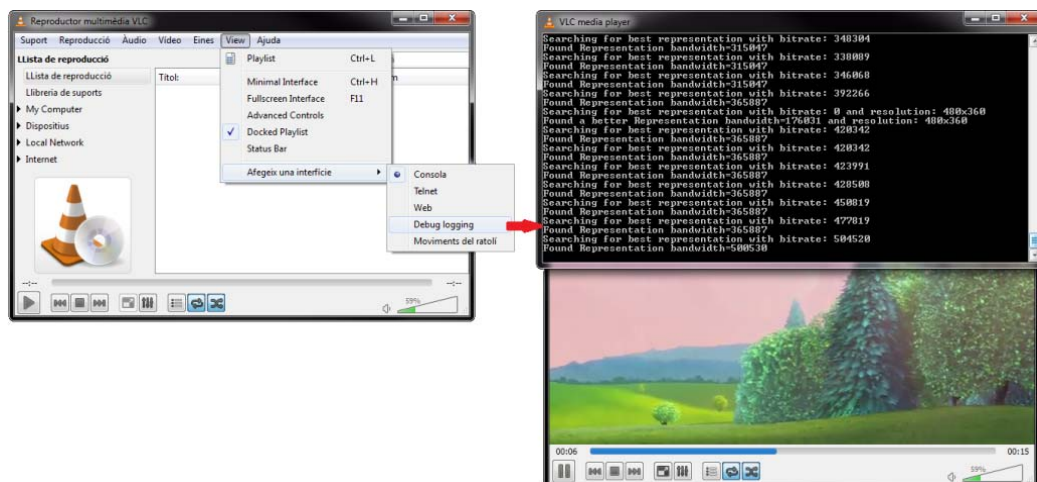


Fig. 3.3 Activant el Log de VLC

També es pot crear un fitxer de log de VLC, tanmateix, aquest fitxer no registra la informació que ens mostra DASH com s'ha indicat amb la opció que s'acaba d'explicar, però sí que podem veure la comunicació inicial que hi ha amb el servidor. Per activar l'opció, **Eines>Preferències** (també amb el llistat avançat)>**Avançat>S'està registrant**.

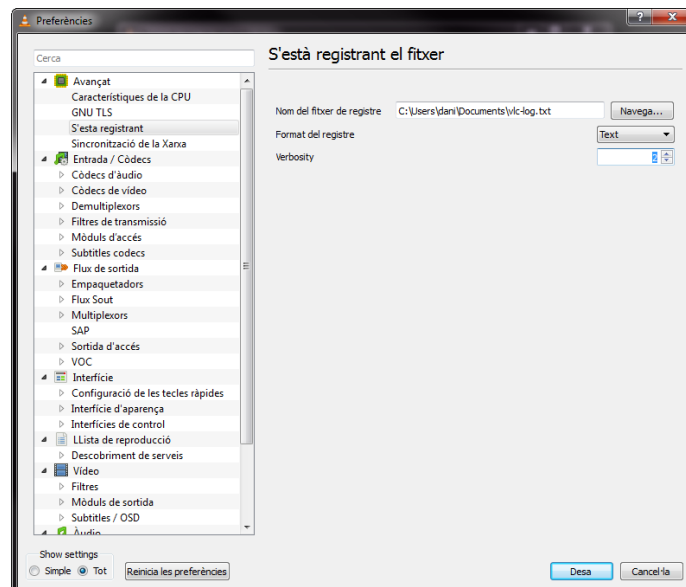


Fig. 3.4 Creant un fitxer de Log

A l'hora de fer proves però, també és interessant arrencar VLC des de la línia de comandes, ja que així podem arrencar totes les anteriors opcions a la vegada sense haver d'anar buscant pels menús. La crida és del tipus:

```
$>vlc.exe --extraintf logger http://servidor/fitxer.mpd
```

Amb aquesta senzilla comanda podem arrencar directament el log que ens permetrà veure en tot moment l'activitat del mòdul DASH.

3.3. MP4Client

3.3.1. Informació general

MP4Client, que es pot trobar a la pàgina de GPAC [13], és una aplicació creada pel grup de GPAC. Aquesta senzilla aplicació no és més que un reproductor, igual que VLC. També és una aplicació que ja existia quan es va començar el treball de final de carrera.

Es pot carregar des de línia de comandes com VLC amb l'arxiu MPD per tal de carregar directament l'stream. Exemple:

```
$>MP4Client.exe http://servidor/fitxer.mpd
```

3.3.2. Avantatges i inconvenients

En aquest treball, el software utilitzat per fer les proves ha estat el VLC, per part del reproductor, principalment perquè era el software oficial. Això ha deixat a

MP4Client retirat a una segona plana. Tot i així, aquest reproductor és funcional, amb els streams preparats per l'MP4Client.

És un reproductor a tenir en compte si continua millorant la part de DASH. De fet una de les mancances que té VLC per la seva pròpia estructura interna, és la fixació d'una resolució per l'stream DASH. En canvi, amb l'MP4Client es pot modificar la resolució i per tant la qualitat, però manualment. Per fer això només cal pujar la qualitat amb Ctrl+H, o baixar-la amb Ctrl+L. Tanmateix, triga bastant en fer aquest canvi.

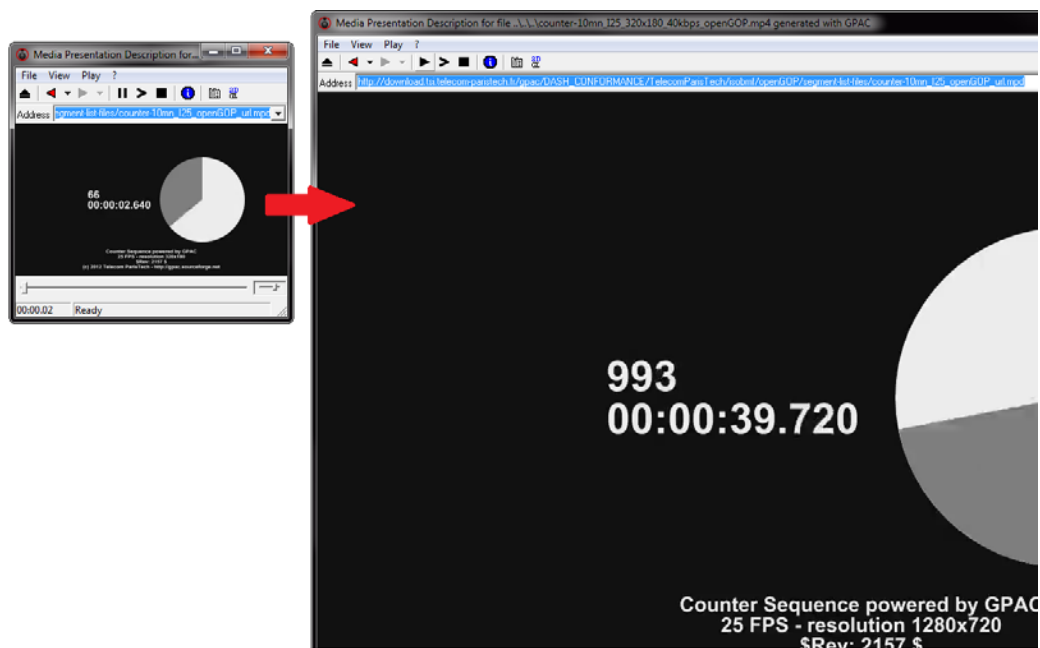


Fig. 3.5 Canvi de resolució amb Mp4Client

Un altre de les diferències és el profile que utilitzen per l'MPD. Mentre que VLC utilitza el profile "urn:mpeg:dash:profile:isoff-main:2011", MP4Client utilitza tant "urn:mpeg:dash:profile:full:2011" com "urn:mpeg:dash:profile:isoff-on-demand:2011" així com també algunes diferències en el format del MPD. Aquestes petites diferències fan que, de moment, els MPDs generats amb format de MP4Client no funcionen en VLC i a la inversa.

3.4. x264

3.4.1. Informació general

X264 [21] és una llibreria que ja porta uns anys en el mercat que permet codificar continguts de vídeo en H.264 / MPEG-4 Part 10.

Aquest software és la referència a l'hora de crear arxius MP4, no només pel grup oficial de DASH, sinó per altres aplicacions. I és que actualment, és un dels software que ofereixen més possibilitats en la codificació de vídeos en MPEG-4.

3.4.2. Utilització de l'x264

Per començar, partim de la base que necessitem un video codificat per poder utilitzar l'MP4Box o, en tot cas, necessitem l'x264 per utilitzar la eina DASHEncoder. Això és important perquè significa que, o bé busquem videos en totes les qualitats i bitrates possibles utilitzant el MP4Box i tenir un sortit acceptable de resolucions i bitrates per tal de fer un stream realment adaptatiu, o bé a partir d'un vídeo *raw* es creen, mitjançant l'x264, totes les resolucions que es vulguin o es considerin suficient per tenir un stream adaptatiu. Al cap i a la fi, quan configurem DASH en el nostre servidor, el que voldrem serà tenir disponibles per l'usuari la major oferta possible de qualitats, resolucions i bit rates, per tant, la segona opció sol ser més adient.

Per aquest motiu la utilització de l'x264 es presenta amb el DASHEncoder. Però tot i així, s'han utilitzat només un parell de paràmetres com la resolució i el bitrate de sortida. Tota la resta de paràmetres de codificació s'han deixat de banda ja que no eren determinants per provar DASH.

3.5. MP4Box

3.5.1. Informació general

Aquesta eina forma part del grup de GPAC, al igual que MP4Client. I si de MP4Client s'ha dit que no s'ha utilitzat gens, no es pot dir el mateix de MP4Box. Per començar, GPAC va ser el primer grup en oferir una solució més que acceptable per generar un stream DASH. Aquest és el motiu pel qual fins i tot el grup oficial de DASH utilitza MP4Box com a eina de generació d'streams.

Com que l'estàndard DASH també accepta entre d'altres MPEG-2 TS (Transport Streams), els desenvolupadors de GPAC també han preparat l'MP4Box per tal que suporti la creació d'streams DASH a partir dels MPEG-2 TS, encara que amb algunes restriccions. Sobre aquesta eina s'han fet diverses proves amb diferents paràmetres que més tard es veuran en aquest treball.

3.5.2. Utilització

MP4Box treballa amb fitxers ISOBMFF principalment i, això significa que, a l'hora de crear un stream DASH, aquest ha d'estar ja codificat. Hem de tenir presents que un fitxer MP4 codificat, amb l'eina x264 per exemple, només el podrem empaquetar de manera que compleixi la normativa DASH, però que aquest video codificat ja té unes característiques definides per la pròpia codificació. Això significa haver de codificar (amb x264 per exemple) l'arxiu *raw* un cop per cada qualitat / ample de banda que desitgem al nostre stream DASH. DASHEncoder ens facilita aquesta tasca.

Com ja s'ha dit abans, aquesta eina té moltes més possibilitats que no pas les utilitzades en DASH. Les esmentades a aquesta taula són només una petita part de les possibilitats que té MP4Box.

Taula 3.1. Comandes de MP4Box

Comanda	Exemple 1	Descripció
-dash X	-dash 2000	Crea fitxers individuals de X mil·lisegons
-frag X	-frag 2000	Utilitza fragments de video (moof) de X mil·lisegons
-segment-name	-segment-name Fitxer	Dona nom a cada segment. En el cas de l'exemple Fitxer1.m4s, Fitxer2.m4s, etc. Si no s'especifica un nom concret, començarà per 1.m4s, 2.m4s, etc.
-rap	-rap	Separa els segments de manera que els paquets de l'inici siguin punts d'accés aleatoris (RAP ⁶).

3.6. DASHEncoder

3.6.1. Informació general

DASHEncoder no es cap codificador, malgrat el seu nom. És tan sols una recopilació d'eines per part del grup oficial DASH de manera que permeten, des d'un arxiu *raw*, passar a tenir tota una sèrie d'streams DASH amb les resolucions, qualitat i velocitats que es vulguin.

Malgrat que no és una eina amb un contingut massa innovador, com ara MP4Box que sí que fa l'encapsulament en format DASH, DASHEncoder aconsegueix d'una manera molt més còmode tenir tot un conjunt DASH.

DASHEncoder utilitza varies eines per arribar al seu objectiu. De moment, utilitza tant video còdecs i àudio còdecs externs, així com també l'encapsulador DASH MP4Box de GPAC. Utilitzar aquesta eina és ben senzill, encara que s'hagi de tenir alguns coneixements de les eines que s'empren; per això les hem mencionat en apartats anteriors.

3.6.2. Utilització

En aquest treball s'ha utilitzat aquest mateix software per la creació de DASH streams. I aquest ha utilitzat el codificador de vídeo x264 i l'encapsulador DASH

⁶ Els RAP, o punts d'accés aleatori, és una eina que s'utilitza en els codificadors de vídeo per marcar un punt important del flux des del qual la reproducció del flux és possible.

MP4Box.A l'hora de realitzar les proves d'aquest treball s'ha obviat l'àudio simplement per la similitud en el tractament en quant a DASH.

DASHEncoder utilitza un fitxer de configuració per poder realitzar la seva tasca. Per veure amb més detall la configuració de DASHEncoder, a l'annex A.III s'ha inclòs un fitxer DASHEncoder.config utilitzat en el treball.

3.7. DASH server

3.7.1. Informació general

A l'hora de crear un servidor per poder oferir continguts DASH, el propi estàndard ho ha deixat molt fàcil. Com és un estàndard basat en HTTP, només s'ha de configurar un servidor HTTP.

En aquest treball hem hagut de configurar servidors HTTP en diverses ocasions, i com que han estat diverses configuracions, al final ha resultat que s'ha treballat amb el servidor Apache i també amb el `httpd`.

Encara que l'Apache no té massa dificultat d'instal·lació i les configuracions que venen per defecte són adequades per les nostres proves, quan s'ha hagut de configurar el `httpd`, ha estat un pel més difícil. En la configuració de `httpd` s'ha de crear un grup d'usuaris perquè un usuari en concret, en aquest cas qualsevol que accedeixi al servidor, pugui entrar a la pàgina web, que en aquest cas seria el nostre arxiu `mpd`.

3.7.2. Configuració

Un aspecte important és que els arxius han d'estar organitzats en carpetes de la mateixa manera que l'arxiu `mpd` indica. De totes formes, GPAC ja ho preveu, i un cop s'ha preparat un video per DASH amb MP4Box, només hem de copiar tota la carpeta que conté els arxius i posar-la al servidor.

També s'ha de tenir especial cura amb els permisos dels arxius. En aquest cas, hem de declarar diferents permisos per cada objecte. A l'hora de crear el sistema d'arxius, tots els fitxers, tant `mpd` com els fitxers de video, han de fixar-se els permisos amb: **`chmod 644 fitxer`**. Mentre que per els directoris, s'utilitza **`chmod 755 directori`**. Com que els servidors solen córrer sota Linux, s'ha utilitzat com a exemple la comanda **`chmod`**. Evidentment, es pot treure l'equivalent pels permisos de Windows segons el tipus d'usuari.

3.8. Dummynet i FrenzyBSD

3.8.1. Informació general

Dummynet [19] serveix per emular una xarxa IP sobre una LAN, introduint fenòmens com ara errors, retard, jitter i limitació d'ample de banda. Aquesta eina està integrada com a mòdul de ipfw (ipfirewall) de FreeBSD. Per això al parlar de Dummynet, s'ha de parlar també de FrenzyBSD.

La distribució lliure FrenzyBSD [22] és una distribució que parteix de la base de FreeBSD. Encara que FrenzyBSD, com a sistema operatiu conté moltes eines diverses, en aquest treball ens hem limitat a utilitzar FrenzyBSD com a servidor HTTP i com a emulador de xarxes amb Dummynet.

3.8.2. Utilització

Dummynet ens permet emular la xarxa a través de *pipes*⁷ que configurarem amb ipfw. D'aquesta manera es pot modelar cada entrada i sortida diferent amb diverses configuracions. De totes maneres, com que els escenaris que utilitzarem seran força senzills, una única *pipe* serà suficient en la majoria dels cassos. L'única excepció serà a l'hora de emular el jitter com ja explicarem.

3.8.3. Pipes

A la figura 3.6 es mostra un dels cassos que veurem en aquest treball.

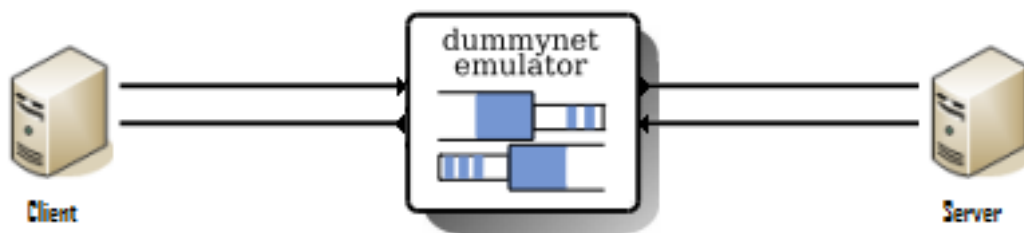


Fig. 3.6 Cas senzill de Dummynet.

En aquest cas per la configuració d'aquesta pipe utilitzarem:

```
$> ipfw add 100 pipe X ip from SERVER to CLIENT
```

on X és un número diferent per cada *pipe* que vulguem configurar, en aquest cas podria tenir el valor 10, per exemple. Evidentment, pretenem veure la resposta dels streams DASH sota certes característiques, per tant no cal crear cap *pipe* "from any to any" o "from CLIENT to SERVER" ja que no és

⁷ Una pipe en Dummynet és el concepte d'un flux unidireccional de dades. Aquest flux de dades pot ser tan genèric com dades IP, o tan específic com el flux de dades TCP que va des d'una màquina A a una màquina B, però no a la inversa.

l'objectiu d'aquest projecte estudiar la resposta d'una sol·licitud HTTP sota condicions adverses, sinó com es comporta DASH sota condicions adverses, en aquest cas, sota el nostre client DASH.

3.8.4. Configuració de Pipes

A partir d'aquí, quan ja tenim creades les *pipes*, podem donar-li la configuració desitjada amb la comanda:

```
$> ipfw pipe X config<OPCIO1><OPCIO2><ETC>
```

Amb aquesta comanda podem escollir varies opcions:

Taula 3.2. Opcions utilitzades amb Dummynet

Opció	Exemple 1	Exemple 2	Descripció
bw <i>SpeedUnits/Sec</i>	bw 2000Kbit/s	bw 10Mbit/s	Limita l'ample de banda del flux que passa per la <i>pipe</i>
delay <i>Timems</i>	delay 3ms	delay 100ms	Obliga a tots els paquets de la <i>pipe</i> a patir un retard
plr value	plr 0.05	plr 0.5	Taxa en tant per un de pèrdua de paquets.

Tot i que hi ha més opcions, com ara *queue* per crear cues, i *mask* per crear màscares de configuració, aquestes opcions no s'han arribat a utilitzar en el treball.

3.8.5. Emular el jitter

Amb les opcions que ens dona una sola pipe podem emular l'ample de banda, el retard i les pèrdues, però encara hi ha un element important en les xarxes que es pot emular de manera diferent: el jitter.

El jitter és un concepte molt important en la commutació de paquets. El concepte es genera quan per arribar des d'un punt a un altre, un paquet pot anar a través de diferents camins. Això provoca que el paquet següent tingui la probabilitat d'escollir un altre camí i, tot i haver sortit després que el primer, arribar abans degut a diversos problemes. El jitter, en definitiva, provoca un retard aleatori individual per cada paquet. En aplicacions multimèdia, que un paquet arribi més tard que un altre pot ser crític segons la importància d'aquests paquet i l'ordre de descodificació d'aquest.

Dummynet no pot reproduir un jitter totalment aleatori, però el que sí pot fer és assignar diferents configuracions per un mateix flux de dades, que succeiran segons una variable aleatòria. És a dir, podem configurar Dummynet per que utilitzi una configuració diferent escollida a l'atzar.

Això s'aconsegueix a l'hora de crear les *pipes* introduint la opció *prob*, que estableix per a cada paquet una certa probabilitat d'anar per una o altra pipe. Cal esmentar que per poder crear les pipes amb la opció *prob*, aquestes pipes han de representar el mateix flux de dades.

```
$> ipfw add prob 0.5 pipe 1 ip from SERVER to CLIENT
$> ipfw add prob 0.5 pipe 2 ip from SERVER to CLIENT
$> ipfw add pipe 3 ip from SERVER to CLIENT
$> ipfw pipe 1 config delay 10ms
$> ipfw pipe 2 config delay 50ms
$> ipfw pipe 3 config delay 1000ms
```

Manca recalcar que al utilitzar *prob* a la primera *pipe* estem aconseguint una probabilitat del 50% de que qualsevol paquet del flux de dades tingui un delay de 10 ms. En canvi, amb el segon *prob*, el que aconseguim és que de la resta del flux de dades, és a dir l'altre 50% que no entra dins del *pipe* 1, el 50% entrarà al *pipe* 2, per tant aconseguim una probabilitat del 25% de que els paquets tinguin delay de 50 ms. Finalment, la resta del flux (un 25% del total) anirà pel *pipe* 3 amb un delay d'1 segon.

3.9. WireShark

3.9.1. Informació general

WireShark [23] és un analitzador de protocols. Aquesta eina permet capturar i analitzar totes les dades que arriben per una interfície de xarxa. Aquesta eina s'ha utilitzat principalment per fer captures de la comunicació entre client i servidor, comprovant la comunicació HTTP i TCP.

3.9.2. Utilització

Utilitzar Wireshark pel que el volem utilitzar en aquest treball es ben fàcil. Un cop obrim el WireShark, anem a **Capture>Interfaces**.

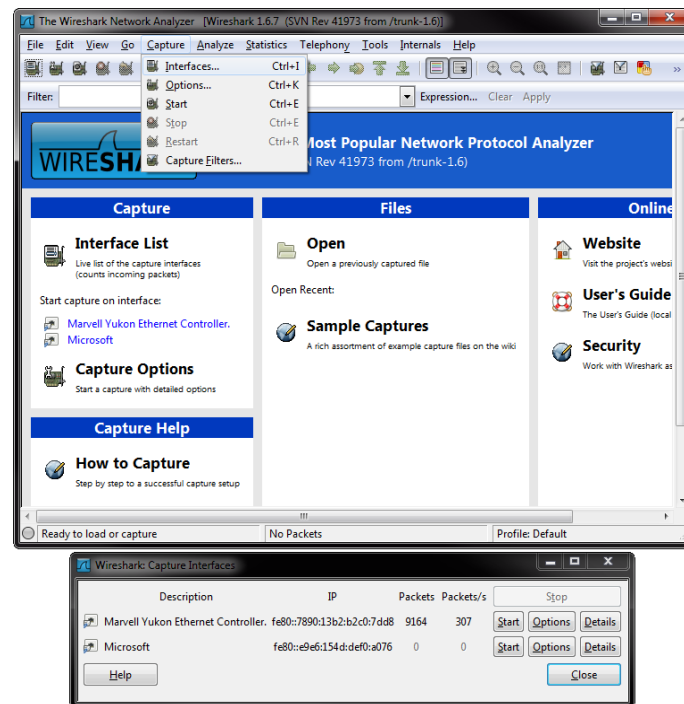


Fig. 3.7 Capturar flux de dades amb Wireshark

Després d'això, aquest software llistarà tot tipus de dades que arribin a la interfície. Com que DASH es un flux de dades HTTP, ens interessarà capturar únicament aquest tipus de dades, per fer això només cal posar "http", a la barra de filtres per poder veure exclusivament el flux de dades HTTP.

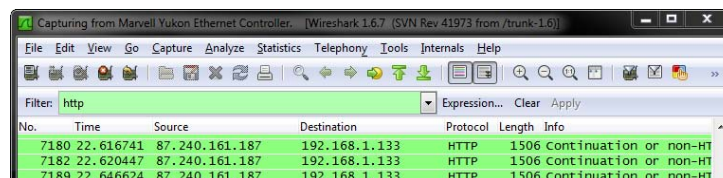


Fig. 3.8 Filtrat del flux per filtre

3.10. MPEG-DASH MPD Validator

3.10.1. Informació general

Aquesta eina, encara que no ha estat imprescindible per realitzar les proves, sí és una eina imprescindible en l'àmbit professional.

[1] En aquest treball ja s'ha parlat dels MPD, arxius XML que han de complir regles sintàctiques per tal que compleixi l'estàndard DASH. Aquests arxius tenen molts paràmetres, i encara que en el treball han estat generats automàticament per un software que és validat pel grup oficial, la majoria vegades es creen amb errors. Errors que possiblement no siguin crítics a de fer les proves, i per això funcionen, però que incompleixen, d'una altre, l'estàndard. L'eina que estem descrivint, que s'ha creat per solventar

aquests petits errors, es pot trobar en una entrada [24] de la pàgina web oficial DASHEncoder, http://www-itec.uni-klu.ac.at/dash/?page_id=282

[2] Llista de correu sobre DASH, <http://lists.uni-klu.ac.at/mailman/listinfo/dash>

[3] LibDASH, http://www-itec.uni-klu.ac.at/dash/?page_id=400

[4] Pàgina web oficial d'Apache, <http://www.apache.org>

[5] Pàgina web oficial de thttpd, <http://www.acme.com/software/thttpd/>

[6] Pàgina web oficial de Dummynet, <http://info.iet.unipi.it/~luigi/dummynet/>

[7] Demostració de canvi de sessió, <http://www.youtube.com/watch?v=f7UG6bfL3gw>

[8] Pàgina web oficial d'x264, <http://www.videolan.org/developers/x264.html>

[9] Pàgina web oficial de FrenzyBSD, <http://frenzy.org.ua/eng/>

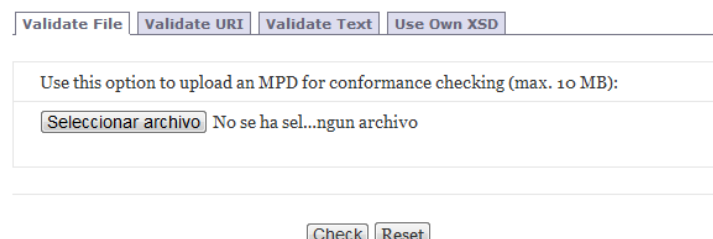
[10] Pàgina web oficial de WireShark, <http://www.wireshark.org/>

.

Per poder fer la validació d'aquests MPD utilitzen uns XML Schema Definition (XSD) determinats. Aquests fitxers són XML genèrics que descriuen el contingut d'un MPD vàlid. En aquest XSD es descriu, la quantitat d'elements, el seu ordre i els valors que tindrà un MPD qualsevol.

3.10.2. Validació de fitxers locals

El mètode per utilitzar l'eina és ben senzill, i amb moltes opcions per donar-nos vàries possibilitats. Aquesta eina ens permet seleccionar un fitxer local, seleccionant un fitxer del nostre ordinador i pujant-lo a la web per poder-lo validar.

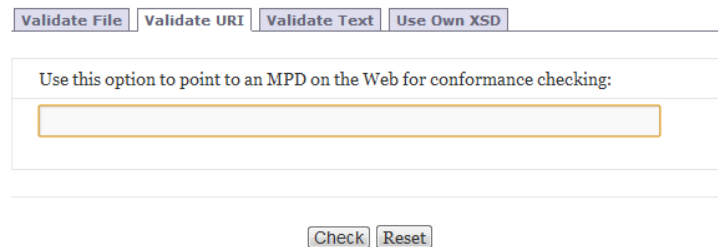


The image shows a web interface for validating MPD files. At the top, there are four tabs: "Validate File", "Validate URI", "Validate Text", and "Use Own XSD". The "Validate File" tab is selected. Below the tabs, there is a text box that says "Use this option to upload an MPD for conformance checking (max. 10 MB):". Underneath this text box is a button labeled "Seleccionar archivo" and a message "No se ha seleccionado ningún archivo". At the bottom of the interface, there are two buttons: "Check" and "Reset".

Fig. 3.9 Validació d'MPD per fitxer

3.10.3. Validació de fitxers en servidor

També ens permet seleccionar una URI, una direcció d'un MPD en la xarxa per poder comprovar si un MPD és correcte o no.



The screenshot shows the 'Validate URI' tab selected. At the top, there are four tabs: 'Validate File', 'Validate URI', 'Validate Text', and 'Use Own XSD'. Below the tabs, a text label reads 'Use this option to point to an MPD on the Web for conformance checking:'. Underneath this label is a single-line text input field. At the bottom of the interface, there are two buttons: 'Check' and 'Reset'.

Fig. 3.10 Validació d'MPD per URI

3.10.4. Validació de text

La última opció és validar directament l'XML en text, copiant el contingut de l'XML al quadre de text que presenta l'eina.



The screenshot shows the 'Validate Text' tab selected. At the top, there are four tabs: 'Validate File', 'Validate URI', 'Validate Text', and 'Use Own XSD'. Below the tabs, a text label reads 'Use this option to directly enter an MPD for conformance checking:'. Underneath this label is a large, multi-line text area for entering XML content. At the bottom of the text area, there are two small buttons: 'Example 1' and 'Example 2'. At the very bottom of the interface, there are two buttons: 'Check' and 'Reset'.

Fig. 3.11 Validació d'MPD per text

3.10.5. Canvi d'esquema de validació

Finalment, una opció extra que han activat és la de poder escollir un esquema de validació propi. Es pot escollir des de XSD remots allotjats en un servidor, com pujar fitxer locals.

The image shows a web-based validation interface with four tabs at the top: 'Validate File', 'Validate URI', 'Validate Text', and 'Use Own XSD'. The 'Use Own XSD' tab is selected. Below the tabs, there is a section titled 'Specify here your own MPEG-DASH XSD which is used for validation (e.g., profiled XSD)'. This section contains two radio buttons: 'Standard XSD' and 'Remote XSD', with 'Remote XSD' being selected. Below these options, there is a text area for entering the XSD path, with instructions: 'Use this option to point to an XSD on the Web or on the computer for conformance checking: Please enter below the path to your XSD. The XSD will be processed after clicking "check" on one of the other registers.' An example section provides paths for Remote, Local (Windows), and Local (Linux). At the bottom of the form is a large, empty text input field.

Validate File Validate URI Validate Text Use Own XSD

Specify here your own MPEG-DASH XSD which is used for validation (e.g., profiled XSD)

☐ Standard XSD

☒ Remote XSD

Use this option to point to an XSD on the Web or on the computer for conformance checking:

Please enter below the path to your XSD. The XSD will be processed after clicking "check" on one of the other registers.

Example:
Remote: `http://www.server.com/DASH.xsd`
Local (Windows): `file:///C:/schemas/DASH.xsd`
Local (Linux): `file:///home/user/schemas/DASH.xsd`

Fig. 3.12 Opció de diferents esquemes XSD

3.10.6. La validació

S'ha de tenir en compte que aquesta eina està impulsada pel grup oficial i, per tant, els MPD fets pel mòdul DASH de VLC funcionen correctament.

Tanmateix, aquesta eina pot no ser útil en alguns casos, ja que si utilitzem MP4Client com a client DASH, els MPDs validats correctament no es carreguen amb MP4Client, en canvi, si es carrega un MPD invàlid (des del punt de vista d'aquesta eina amb l'XSD oficial), o el que és el mateix, preparat per MP4Client, aquest reproductor és capaç de llegir l'MPD i reproduir l'stream DASH. Això pot ser degut a un XSD DASH ja obsolet.

The figure shows two side-by-side validation interfaces for DASH-MPD files. Both interfaces have tabs for 'Validate File', 'Validate URI', 'Validate Text', and 'Use Own XSD'. The left interface (GPAC) shows a validation failure, while the right interface (VLC) shows a validation success.

Left Interface (GPAC):

- URL: `http://download.tsi.telecom-paristech.fr/gpac/DASH_CONFORMANCE/Tele`
- Buttons: Check, Reset
- Result: **Your DASH-MPD is NOT VALID** (Red box with a red X)
- XSD for checking: DASH International Standard XSD
- Start XLink resolving: =====
- XLink resolving successful (Green checkmark)
- Start MPD validation: =====
- Errors:
 - Line:Col[14:38]:cvc-complex-type.3.2.2: Attribute 'subsegmentStartsWithSAP' is not allowed to appear in element 'Representation'.
 - Line:Col[15:59]:cvc-complex-type.2.4.a: Invalid content was found starting with element 'ContentComponent'. One of '{ "urn:mpeg:DASH:schema:MPD:2011":FramePacking, "urn:mpeg:DASH:schema:MPD:2011":AudioChannelConfiguration, "urn:mpeg:DASH:schema:MPD:2011":ContentProtection, WC[##other:"urn:mpeg:DASH:schema:MPD:2011"], "urn:mpeg:DASH:schema:MPD:2011":BaseURL, "urn:mpeg:DASH:schema:MPD:2011":SubRepresentation, "urn:mpeg:DASH:schema:MPD:2011":SegmentBase,

Right Interface (VLC):

- URL: `http://www-itec.uni-klu.ac.at/ftp/datasets/mmsys12/BigBuckBunny/MPDs/B`
- Buttons: Check, Reset
- Result: **Your DASH-MPD is VALID** (Green box with a green checkmark)
- XSD for checking: DASH International Standard XSD
- Start XLink resolving: =====
- XLink resolving successful (Green checkmark)
- Start MPD validation: =====
- MPD validation successful – DASH is valid! (Green checkmark)
- Start Schematron validation: =====
- Schematron validation successful – DASH is valid! (Green checkmark)

Fig. 3.13 Comparació entre validacions d'MPD

A la figura 3.13 es mostra, a l'esquerra, un MPD generat per l'MP4Client per part de GPAC. El resultat diu que aquest MPD no és vàlid, malgrat que aquest mateix stream hagi carregat bé en aquest reproductor. A la dreta podem observar el resultat vàlid d'un MPD generat per l'VLC.

Com a conclusió, aquesta eina ens pot servir per corregir errors bàsics com no col·locar bé el Base-URL apuntant a un arxiu local en comptes del nostre servidor, o ens pot servir per assegurar-nos que li hem donat una duració determinada a l'stream, però aquesta és només una eina de validació sintàctica, i és possible que, en certs reproductors, no sigui necessari que es verifiqui al 100% l'estàndard.

3.11. LibDASH

3.11.1. Informació general

LibDASH [16] és una llibreria implementada en .NET framework, concretament en C++. I per tant, a diferència de les altres eines que hem nombrat, aquesta és exclusiva per Windows. Encara que bé hi ha eines per adaptar .NET a distribucions Linux, la llibreria pot no funcionar en aquests entorns.

3.11.2. Utilització

Aquesta llibreria neix amb la intenció de donar un suport per projectes que utilitzin la tecnologia DASH. Amb aquest objectiu han creat una llibreria capaç de crear un escenari DASH i carregar l'stream en només quatre passos, tal com es mostra a la figura 3.15.

```
IDASHManager *manager = CreateDashManager();
manager->open("http://...mpd");
manager->setBufferLength(sec);
manager->read(pBuffer, length);
```

Fig. 3.14 Exemple d'ús de libDASH

D'aquesta manera tan senzilla comencem a omplir el nostre buffer amb el vídeo DASH.

3.11.3. Valoració

Encara que és molt senzill obrir un fitxer DASH i començar a llegir, per poder controlar paràmetres s'ha de ficar una mica més en el codi. Però tampoc és massa complicat dedicant-hi una mica de temps, ja que la llibreria està oberta completament, així que observant el codi de la classe DASHManager, s'aconsegueix esbrinar per què utilitza cada classe i així poder veure que s'ha de fer.

Sense entrar massa en detall, i malgrat que sigui tot obert i accessible, les conclusions que traiem són que el sistema podria ser una mica encara més flexible. Realment, amb aquesta llibreria no tens prou llibertat de moviments per poder escollir res directa i fàcilment: ni resolucions, ni temps de buffer, ni qualitat. En aquest sentit podria deixar-li una mica més de possibilitats al client de la llibreria. Al cap i a la fi, és el client el que determina, amb les possibilitats que té, una resolució major o menor, o un buffer més o menys gran segons la memòria que tingui o es vulgui utilitzar. Si es vol controlar tot això, s'ha de profunditzar una mica més en la llibreria.

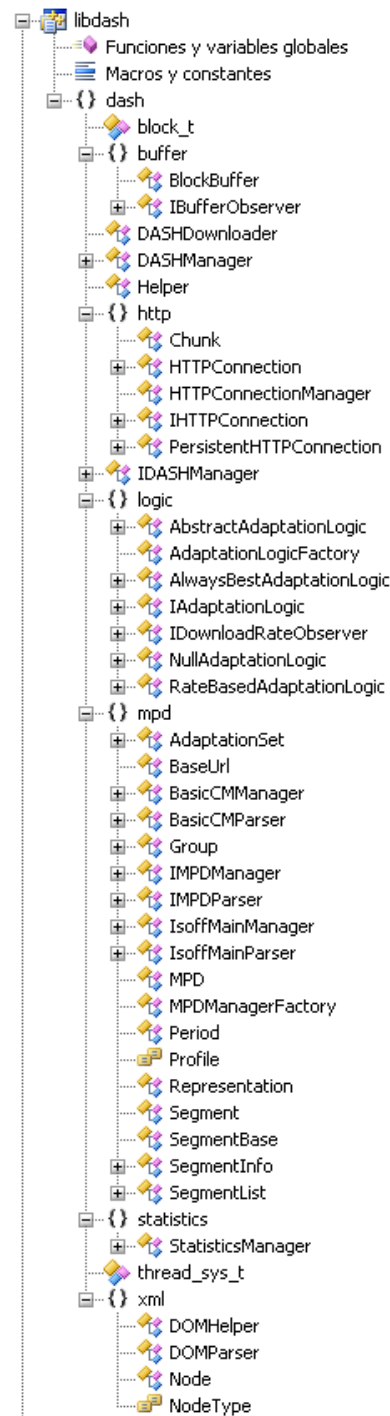


Fig. 3.15 Objectes de libDASH

3.12. DASH-JS i Google Chrome

3.12.1. Informació general

Google Chrome és un altre client DASH en una fase molt inicial de proves. Utilitzant el mòdul javascript (JS) adient i la configuració bàsica, és converteix en un reproductor DASH.

3.12.2. Configuració de Google Chrome per DASH

DASH-JS [25] és una llibreria javascript que neix amb la finalitat de possibilitar a tots els vídeos de la xarxa implementats en HTML5 l'ús de DASH. Aquesta eina s'ha fet sota les proves de l'únic navegador que l'utilitza: Google Chrome a partir de la versió 17. Però ara per ara, aquesta eina no és més que una característica en procés d'avaluació i per tant, actualment s'han d'activar unes característiques determinades. Per fer això, cal anar a la barra de direccions de Google Chrome i introduir:

chrome://flags

Veurem una pàgina amb diverses característiques, la majoria desactivades. Busquem la característica anomenada “API de fonts de medis en els elements de <video>”. Després d'activar-la és important reiniciar l'aplicació amb el botó que apareixerà més avall.

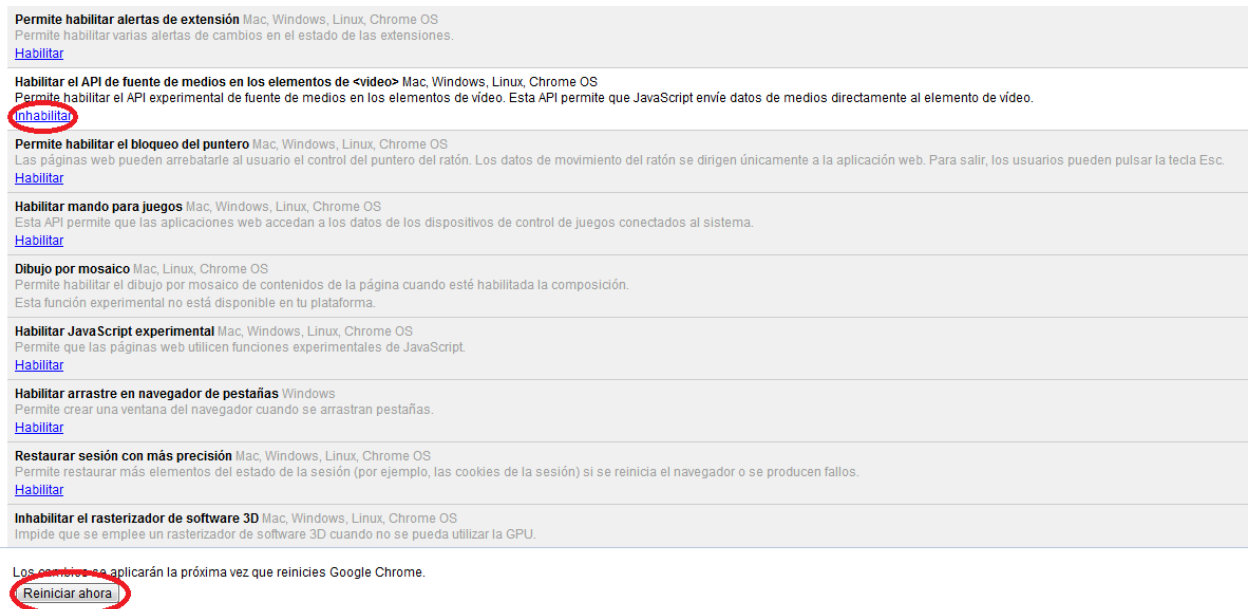


Fig. 3.16 Habilitar DASH-JS en Google Chrome

3.12.3. Utilització

[11] Amb aquesta característica activada podem anar a la pàgina web que ens marca la guia oficial per poder veure el nostre stream DASH en HTML5

Aquesta eina ens permet veure un stream DASH mitjançant un senzill script. Iniciem la reproducció del vídeo i un gràfic que ens mostrarà l'ample de banda utilitzat amb molt poques línies. Només cal enllaçar els scripts de la llibreria i introduir aquest script final.

```
<script>
  video = document.querySelector('video');
  vardashPlayer = newDASHPlayer(video,"http://www-itec.uni-
    klu.ac.at/dash/js/content/sintel_multi_rep.mpd");
</script>
```

Amb aquest petit script podem visualitzar un vídeo en DASH:

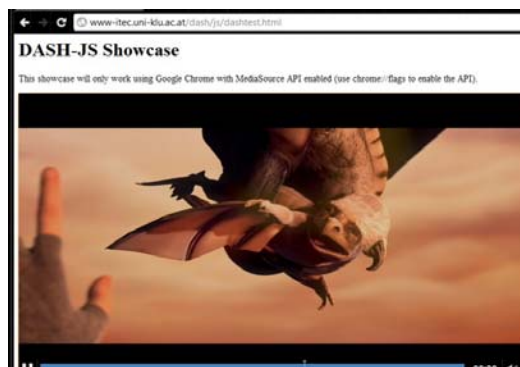


Fig. 3.17 Exemple de reproducció DASH sobre HTML5

3.12.4. Novetats

Tot i que no s'han pogut fer totes les proves desitjades per manca de temps, ha estat interessant utilitzar aquesta eina ja que, a diferència de la nova versió de VLC, aquesta eina ens mostra el gràfic de l'ample de banda consumit en tot moment. Al costat o sota el vídeo, depenent de la resolució del vídeo i de la pantalla de l'usuari, apareixerà un gràfic. A la figura 3.18 podem trobar en blau l'ample de banda consumit i en vermell l'ample de banda disponible.



Fig. 3.18 Ample de Banda

3.13. Mozilla Firefox

3.13.1. Informació general

Mozilla també s'incorpora al grup d'aplicacions que s'incorporen a l'estàndard DASH. El 23 de maig de 2012, dos mesos després que DASH aparegués com una possibilitat a Google Chrome, surt una notícia amb Mozilla [27], però desenvolupat sense Javascript. Aquest cop s'utilitza libDASH, una de les eines mostrades, per avançar en la mateixa direcció. Mozilla reutilitza la llibreria libDASH per incorporar DASH al navegador Firefox.

3.13.2. Activació

Per activar la tecnologia DASH, abans de res cal baixar-ne la *nightly build* que ho permet i després activar aquesta opció mitjançant la configuració de Mozilla, molt similar a com es fa amb Google Chrome.

Abans de res cal introduir a la barra de direcció de Mozilla "**about:config**" per accedir a la configuració, i després, tal i com indica a la figura, s'ha d'introduir una línia pressionant el botó dret del ratolí sobre qualsevol línia. Seleccionem

New>Boolean, aquest ens demanarà un nom per la variable, i s'haurà d'introduir **"media.dash.enabled"** i finalment configurem la variable amb valor *true*, i ja tindrem DASH activat.

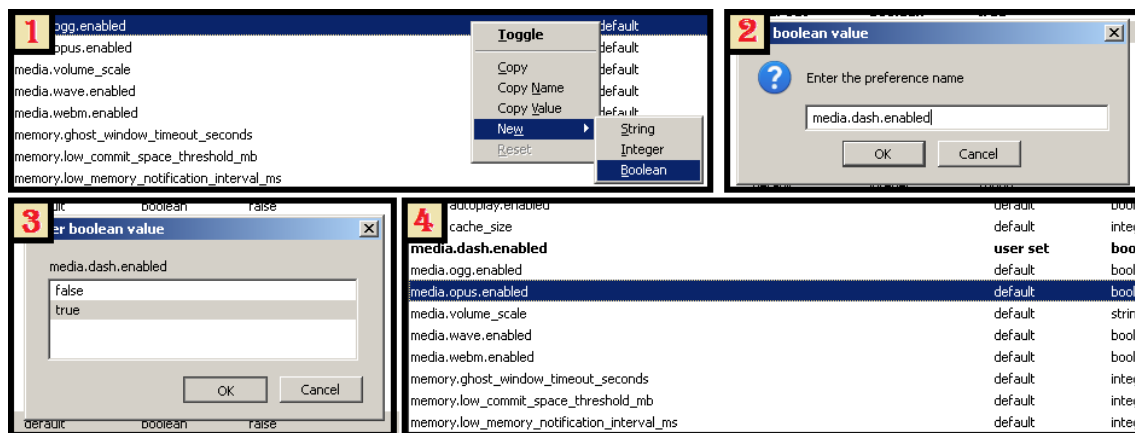


Fig. 3.19 Activació de DASH en Firefox

3.13.3. Utilització

Tot i que ha sortit molt més tard, l'avantatge de Mozilla davant de Google Chrome i DASH-JS, és la facilitat d'ús. Si per utilitzar DASH a Google Chrome s'havia d'enllaçar amb els fitxers scripts i finalment arrencar un script final per que arrenqués el video, amb Mozilla tot queda reduït a la crida de l'etiqueta de <video> a una direcció determinada.

```
<html>
  <head>
    <title>
      DASH
    </title>
  </head>
  <body>
    <video preload="auto" width="852" height="480" controls="controls" src="MPD"/>
  </body>
</html>
```

CAPÍTOL 4. PROVES REALITZADES

L'objectiu d'aquest treball de fi de carrera era configurar un testbed en el qual poder fer proves sobre DASH i/o MMT. Malauradament, com ja s'ha comentat, amb MMT tot just es comencen a aprovar els estàndards de manera que en el moment de fer aquest treball no hi havia cap eina disponible per poder comprovar el funcionament. Per tant, aquest treball s'ha centrat en fer diverses proves amb les eines disponibles per DASH.

En aquest àmbit, s'han pogut muntar diversos escenaris i s'han pogut fer diverses proves amb les eines que hi ha actualment. Aquests escenaris seran els que s'explicaran al llarg d'aquest capítol. Com a resum ràpid, es va muntar un escenari HTTP senzill on poder comprovar les eines. Les eines MP4Client, MP4Box, el mòdul DASH per VLC i el DASHEncoder entre d'altres, que són els visualitzadors i generadors d'streams DASH, es van provar apart abans de fer cap prova en un escenari de proves amb servidor i Dummynet.

Més tard també es va provar introduint una estació Dummynet de manera que es pogués observar la resposta de DASH a diversos problemes de xarxa. Al cap i a la fi, això mateix és pel que s'ha desenvolupat aquest estàndard i s'han de fer proves per veure el correcte funcionament de DASH.

4.1. Instal·lació bàsica

4.1.1. Compilacions i patches de VLC

Després de tota la documentació llegida, li va tocar el torn a les primeres proves. La instal·lació del VLC amb el corresponent mòdul de DASH tal i com es proposava a la pàgina web oficial va ser el primer pas.

S'havia de compilar VLC incloent-hi el mòdul que el grup oficial de DASH havia preparat per fer les primeres proves. L'estudiant va ser incapaç de compilar-lo correctament, va provar tot un conjunt de coses per compilar aquestes dues coses. De les primeres proves que es van fer ser tal i com deien a la pàgina web, aplicar un patch. El patch no funcionava i donava errors, així que es va provar de compilar directament i tocar els diferents arxius del modules.ac i els makefile per mirar de fer-lo funcionar. Es va provar instal·lant altres versions de la distribució Ubuntu, i netejant-la del tot, no fos algun tema d'alguna llibreria mal actualitzada. Al final va resultar impossible.

4.1.2. VLC 2.0

El més irònic es que poc després, una o dues setmanes, de realitzar aquestes proves van llençar el VLC 2.0 i amb aquest, el mòdul de DASH ja venia incorporat. Així que es va a començar a baixar i a instal·lar de seguida. Com

que fins i tot amb el VLC 2.0 van haver diferents problemes, és interessant llistar aquí els passos per si algun lector es troba algun dia res semblant.

1. Compilar a mà.
2. Amb ./bootstrap apareix l'error:
 - a. Configure: error: no package 'speexdsp' found
3. apt-get install speexdsp
 - a. no existeix
4. Es busca per speexdsp: Es troba dins el paquet speex
 - a. URL: <http://downloads.us.xiph.org/releases/speex/speex-1.2rc1.tar.gz>
5. Es compila
 - a. Error 2.a
6. Es torna a compilar amb permisos de superusuari
 - a. Error 2.a
7. S'instal·la mitjançant gestor de paquets d'Ubuntu. Error:
 - a. Configure: error: no package 'XCB Composite' found
8. S'instal·la mitjançant gestor de paquets d'Ubuntu. OK.

Aquest és només un breu resum del que va passar, si es vol més detall es pot entrar al blog del projecte [28] per veure com va evolucionar realment aquest procés. Tanmateix, la conclusió d'això es evident. Sempre que es pugui s'ha d'intentar primer fer tots els processos amb les eines que el propi sistema operatiu ens dona.

4.2. Muntant l'escenari Dummynet

4.2.1. Informació general

A partir d'aquí, es va decidir muntar l'escenari del primer testbed a casa de l'autor del treball. En aquells moment l'estudiant tenia instal·lat un Ubuntu 11.10 i segons la informació cercada per Internet, Dummynet no aconseguia compilar amb aquesta versió. Però si que funcionava amb la versió 9.04.

4.2.2. Intent d'instal·lació

La instal·lació d'aquesta opció, FrenzyBSD, tampoc va ser gens senzill, ja que l'ordinador que va aconseguir l'alumne tenia poca RAM. Així que va intentar emular el FrenzyBSD des d'una màquina virtual amb VirtualBox. I encara que es van fer diverses proves, no es va poder instal·lar Dummynet a la màquina virtual. Només es podia arrencar el disc Live del FrenzyBSD.

No va funcionar durant gaire temps. El fet que aconseguís un PC portàtil amb capacitats limitades, va fer que es descartessin la màquina virtual i es comencés a treballar sobre el portàtil. Era evident, si no hi havia una

instal·lació, no es podien guardar arxius i, per tant, no es podria muntar l'HTTP o s'hauria de muntar cada cop que s'encengués el PC, cosa que no era òptima.

4.3. Primeres proves

4.3.1. Primers resultats

Mentrestant, l'alumne ja tenia instal·lat VLC i va començar a fer proves. El primer resultat va ser nefast, l'stream estava lluny de funcionar correctament. No va ser fins que dies després, va veure que hi havia una opció de caching al VLC, és a dir, de crear un buffer que permetia, entre d'altres, capturar d'una estona els paquets per poder absorbir pèrdues de paquets, per poder enviar abans els paquets més importants dels quals altres paquets depenen o inclús la reordenació de paquets perduts i reenviats.

4.3.2. Caching

Es van fer varies proves amb aquest caching. Els resultats que va extreure llavors van ser molt diferents. Aquí es mostra una taula resum amb dades que va anotar l'estudiant sobre la qualitat del vídeo, amb un stream allotjat en un enllaç [29] ja no disponible. Aquest vídeo era un vídeo de cinc minuts del recorregut d'una mountain bike, un vídeo amb força moviment i canvis d'escena, el que generava canvis de taxa molt freqüents.

Taula 4.1 Resum de la primera prova sobre VLC

Caching	Anotacions		
	Audio	Interrupcions de vídeo	Retard a la recepció
1000 ms(per defecte)	Pèrdua total d'audio	Interrupcions constants	4 segons
10000 ms	Pèrdua des del minut 4	7 interrupcions menors	12 segons
15000 ms	Pèrdua des del minut 4	Sense interrupcions	19 segons
20000 ms	Pèrdua des del minut 4	Sense interrupcions	26 segons

Com es pot veure, les opcions de retard a la recepció estan relacionats amb el caching ja que obtenim un retard d'uns segons superior al caching configurat, un resultat coherent, ja que precisament és el que fa el caching, retardar la recepció per tal de tenir una cua amb suficient informació com per que no hi hagi talls en la reproducció.

Les interrupcions en el vídeo mostren una millora quantitativa a partir dels 10 segons de cache. Però tot i així la pèrdua d'àudio a partir del minut quatre deixava molt que desitjar. I no només això, sinó que el caching amb DASH hauria de ser automàtic. És a dir, una de les "promeses" de DASH era que un video s'adaptés a aquest buffering que necessiten els videos. Mitjançant una pitjor qualitat durant els primers segons, DASH pretén millorar la qualitat

progressivament de manera que aquest buffering sigui transparent a l'usuari, o com a mínim, aquesta era una de les idees que acompanyaven DASH. Això no es veu en aquestes primeres proves.

4.4. Proves amb servidors

4.4.1. Descàrrega de fitxers precodificats

Després dels primers intents, el que tocava era començar a pensar en fer el primer testbed. Es van descarregar de la pàgina web oficial [4] els MPD i streams DASH que hi havia, però el volum d'informació era tan gran, mesurada en molts GBytes, que es va decidir baixar només alguna part per fer proves. Així les primeres proves es van fer sobre VLC, amb uns streams de la pel·lícula Big Buck Bunny com els trobats a [9] i [10]. Primer es van descarregar tots els arxius necessaris per la reproducció dels streams, és a dir, els arxius M4S (segments de vídeo), i els arxius MPD però, tal i com s'ha dit abans, només aquells streams segmentats a 1segon.

4.4.2. Logging amb VLC

Després de fer les proves inicials, es va investigar com fer el logging de VLC. Després de molt buscar es va arribar a diverses conclusions errònies que més tard es resoldrien.

Per començar, encara que es van buscar formes de fer un log interessant, l'únic que es va trobar llavors va ser la comanda:

```
--extraintf http:logger
```

A partir d'aquí, l'estudiant va pensar que només amb aquesta comanda podria veure algun log. Això no és així, més tard descobriria que es pot especificar d'altres maneres, i que només el "logger" és necessari. A l'apartat 3.2 està descrit de manera gràfica el que és necessari per veure el log.

Això permetia veure una pantalla amb l'evolució de l'ample de banda que consumia l'stream i al que s'intentava arribar. De fet, amb aquesta pantalla es van fer moltes proves diverses durant tota la durada del projecte, així com Wireshark amb el qual també es van fer algunes captures.

4.4.3. Desxifrant informació de logging

A la figura 4.1 es pot veure l'activitat del mòdul DASH. Però erròniament es va arribar a la conclusió que el bitrate només pujava, no baixava mai. I això va ser per varies raons. La primera i més important, és que amb les proves que es van fer llavors, encara no s'havia posat el Dummynet, i encara que això ja ho tenia

present l'alumne, no entenia el creixement sense parar d'aquest número. Van ser unes sessions força confuses.

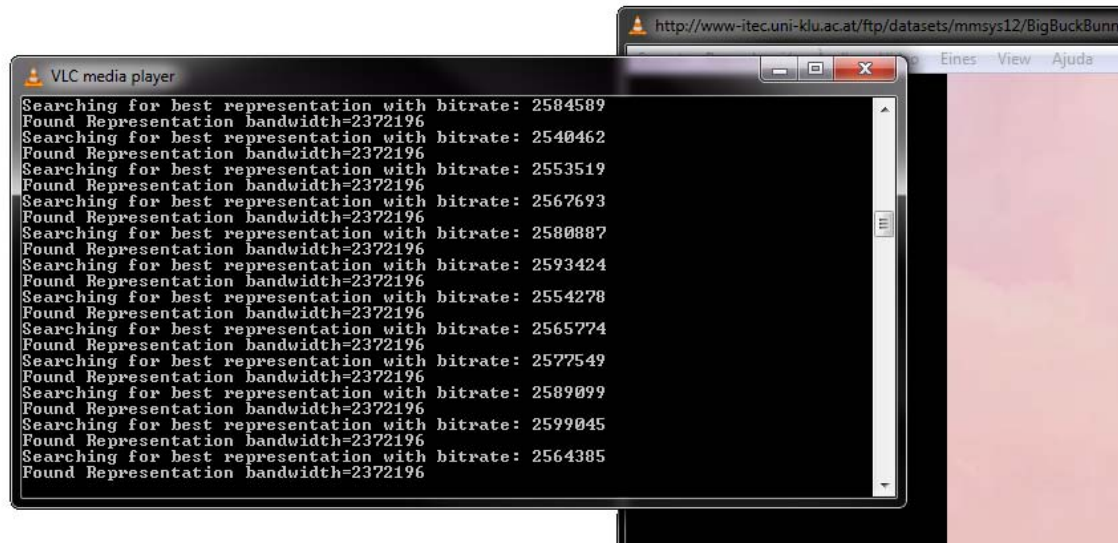


Fig. 4.1 Finestra de logging de VLC

Aquest número representat a la figura 4.1 és el bitrate mesurat en bps (bits/s) que estem aconseguint amb aquesta connexió i vídeo. Observem que el vídeo utilitza un stream que necessita 2372196 bps, mentre que el bitrate observat és del rang de 2500000 bps. Cal tenir present que VLC no permet canviar de resolució a mitja sessió, per tant, com només hi ha dos possibles amplituds de banda per cada resolució amb l'stream oficial, no es pot veure tan clarament aquest canvi degut al ràpid canvi que es fa a l'inici augmentant ample de banda. Tot i que forçant amb Dummynet la connexió es veuria, més tard, aquesta relació.

4.4.4. Altres proves

Després de veure la informació d'VLC, es van instal·lar les aplicacions de GPAC [13] sota Windows, ja que fins ara s'havia treballat amb VLC sota Ubuntu. I es van fer també algunes proves, per veure que no funcionava. Un mes més tard l'alumne es va adonar que no funcionava amb el dataset del Big Buck Bunny, però sí amb streams preparats per aquest reproductor.

4.5. Creant els DASH stream

4.5.1. Preparant l'escenari

La generació d'streams però, va ser bastant més productiva. Amb el descobriment de MP4Box i DASHEncoder es van començar a fer proves amb aquestes eines. I després es va provar aquest software amb VLC i MP4Client.

Per això, es va intentar descarregar l'arxiu *raw* del Big Buck Bunny, però la mida de l'arxiu va fer que l'alumne es decantés per un altre arxiu *raw* més petit. Es va decidir agafar un de curta durada, uns 16 segons, per que no ocupés massa espai de la pàgina xiph **¡Error! No se encuentra el origen de la referencia..**

4.5.2. MP4Box

Amb MP4Box es van fer unes sis proves diferents provant els paràmetres més bàsics. Però abans, es va codificar el fitxer *raw* a 1080p (1920 x 1080 píxels, progressiu) per tal que MP4Box pogués acceptar-ho com a fitxer vàlid.

```
MP4Box -frag 10000 -dash 5000 ../Escritorio/snow_mnt_1080p.mp4
MP4Box -frag 5000 -dash 6000 -segment-name ../Escritorio/ExMP4Box/Ex2/
../Escritorio/snow_mnt_1080p.mp4
MP4Box -frag 10000 -dash 10000 -rap -segment-name ../Escritorio/ExMP4Box/Ex3/
../Escritorio/snow_mnt_1080p.mp4
MP4Box -frag 10000 -dash 10000 -rap -segment-name ../Escritorio/ExMP4Box/Ex4/snow -
frags-per-sidx 3 ../Escritorio/snow_mnt_1080p.mp4
MP4Box -frag 10000 -dash 10000 -rap -segment-name ../Escritorio/ExMP4Box/Ex5/ -daisy-
chain -frags-per-sidx 3 ../Escritorio/snow_mnt_1080p.mp4
```

El primer intent, sense el paràmetre `--segment-name` no produïa els fitxers m4s desitjats, així que es va desestimar. A partir d'aquí totes les comandes són funcionals. Amb la segona línia simplement es creen fitxers de 6 segons cadascun. La tercera comanda permet, a més, que tots els fitxers, encara que no durin exactament 6 segons, estiguin marcats com un RAP, i per tant es pugui reproduir l'stream a partir de qualsevol d'ells, ja que disposaran de les capçaleres necessàries per permetre-ho. La penúltima línia permet crear per cada fitxer, diferents fragments interiorment, per tal de, si tenim segments massa grans, per exemple de 10 segons, d'aconseguir més punts d'accés.

4.5.3. DASHEncoder

La part de DASHEncoder és encara més útil. DASHEncoder funciona amb un fitxer de configuració. Des d'aquest fitxer s'introdueixen els paràmetres de configuració de còdecs i els paràmetres de configuració de l'MP4Box i es comença a crear, des del fitxer *raw* les diferents carpetes amb diferents resolucions i amplituds de banda.

Encara que el fitxer de configuració no es presenta en aquest apartat per problemes d'espai, sí que es pot observar en l'Annex A.III.1. A més, val la pena esmentar aquelles opcions més utilitzades en el DASHEncoder, que són:

Taula 4.2 Opcions de DASHEncoder

input-res	Resolució del fitxer <i>raw</i>
bit-rate	Taxa binària generada. Es pot indicar no només més d'un bit-rate de sortida, amb el que es tindria

el mateix stream a diferents bit-rates, sinó també amb diferents resolucions. Exemples:

bit-rate : 200|400|600|1000

bit-rate : 50@352x288|100@352x288

rap-aligned	Cal activar-ho si es volen els fitxers amb l'opció rap com al MP4Box.
mpd-name	Nom que es donarà al MPD.
url-root	Defineix l'URL base. Molt important a l'hora de moure cap al servidor. Exemple: url-root : http://192.162.1.1/
set-base-url	Cal activar-ho si volem que l'mpd apunti a una URL relativa
mpdActStandardPostfix	Amb aquesta opció l'MPD es torna a crear seguint un patró estàndard
duration	Cal indicar la duració en el format adequat: duration : 0H0M21.54
minBufferTime	Amb aquest indiquem el Buffer mínim
segDuration	Finalment l'últim paràmetre serveix per determinar la durada en temps de cada fitxer.

També cal esmentar que després de diferents intents amb la opció de codificació en 2 passos del x264 que consisteix en codificar un video 2 vegades, agafant informació útil per la segona codificació dels resultats de la primera, donava l'error: "Cannot configure x264 to make 2-pass encoding". Per la resta, es pot dir que ha funcionat com s'esperava.

4.6. Utilitzant Dummynet

4.6.1. Preparant l'escenari

Les proves del cinquè mes i començament del sisè també van ser amb Dummynet. Utilitzant Dummynet per modificar, entre d'altres, l'ample de banda disponible, i creant conjunt d'streams que fossin adequats per fer les proves amb Dummynet, l'estudiant va poder provar finalment amb streams generats per ell de manera que pogués veure correctament la resposta de VLC davant de les limitacions del canal que permet introduir l'emulador (reducció d'ample de banda, o pèrdues).

Per això cal esmentar que el testbed es va fer només amb dos PCs. Mentre que un tenia Windows o Ubuntu, és igual quin dels dos, amb VLC 2.1, l'altre tenia instal·lat un FrenzyBSD amb el servidor thttpd. Aquest últim ens servia

tant de servidor DASH com de emulador de xarxa amb el Dummynet. Per fer les proves s'havia de configurar la targeta de xarxa de l'ordinador amb VLC amb la porta d'enllaç fixada al servidor, ja que amb aquesta topologia, es passava a través d'un router que feia de pont i s'ha d'evitar que les connexions passin a través del router cap a internet.



Fig. 4.2 Configuració del primer testbed

4.6.2. Comprovant el funcionament de DASH

I així es van fer proves, inicialment sense Dummynet, provant la funcionalitat correcta de VLC i que tot s'hagués fet correctament. També observàvem amb el logger que amb un funcionament normal, VLC arrencava DASH amb el mínim de ample de banda que podia per la resolució indicada, i durant la transmissió anava pujant l'ample de banda requerit, per veure si en trobava algun stream amb millor qualitat.

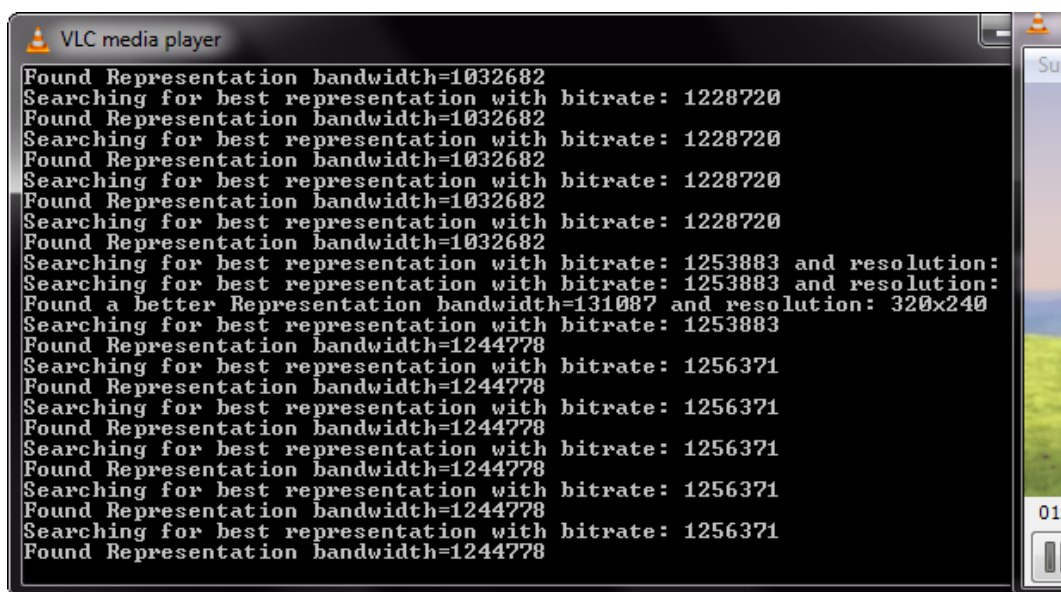


Fig. 4.3 Funcionament normal de DASH

4.6.3. Preparant un altre escenari

Després d'aquestes primeres proves on es prova que el funcionament normal és correcte, es van començar les proves amb FrenzyBSD. Es va poder reproduir un altre cop el mateix, aquest cop amb un servidor HTTP dedicat i un altre FrenzyBSD dedicat, sense routers pel mig, i tots connectats entre ells.

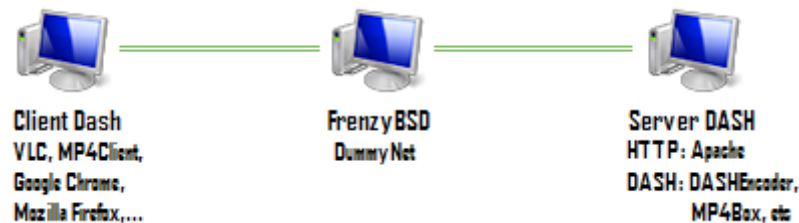


Fig. 4.4 Configuració del segon testbed

4.6.4. Noves instal·lacions

Val la pena esmentar que l'autor del treball va tenir alguns problemes amb la configuració de FrenzyBSD. No se'n va adonar que FrenzyBSD té tot configurat de manera que només cal introduir les regles de les *pipe* Dummynet i començar a fer proves. També, acostumat com estava d'haver estat treballant amb *thttpd*, la configuració d'Apache va resultar tenir alguns entrebancs.

Tot i així, es va muntar correctament un escenari on es van poder fer proves. Però per poder fer proves correctament, l'stream oficial donava una sèrie d'amples de banda per diferents resolucions. El problema és que encara no podia canviar de resolució en mig d'una reproducció. Això fa que l'estudiant es muntés pel seu compte, utilitzant DASHEncoder, un nou stream, l'MPD del qual es pot trobar a l'annex A.II.

L'MPD el va crear l'estudiant per poder simular amb comoditat els canvis de stream per part de VLC. Aquest descriu l'stream d'un trailer de 52 segons de duració. Com que VLC no permet el canvi entre resolucions enmig de la reproducció, aquest MPD només consta d'una única resolució: 1920x1080. Contràriament a l'MPD, on només hi havia 3 o 4 bitrates per cada resolució. En aquest MPD tenim els bitrates (en Kbps): 200, 400, 600, 1000, 1500, 3000, 4500, 6000, 8000 i 10000. Encara que aquest stream descriu un stream amb fitxers segmentats d'1 segon cadascun, també s'han fet proves amb MPDs que descriuen un sol fitxer indexat (amb la mateixa durada de fragment).

4.6.5. Proves amb l'ample de banda

S'han fet diverses proves, utilitzant les eines que proporciona Dummynet amb les pipes, entre les quals, les més abundants han estat les proves amb l'ample de banda.

Per començar, es va provar introduint diferents amplex de banda. Comencem amb un ample de banda de 20000 kbps que supera, per molt, el major ample de banda que podem trobar a l'MPD generat. El resultat és l'esperat, l'ample de banda de seguida arriba al punt on el paràmetre *bandwidth* i un dels paràmetres del logging de VLC coincideixen encertadament.

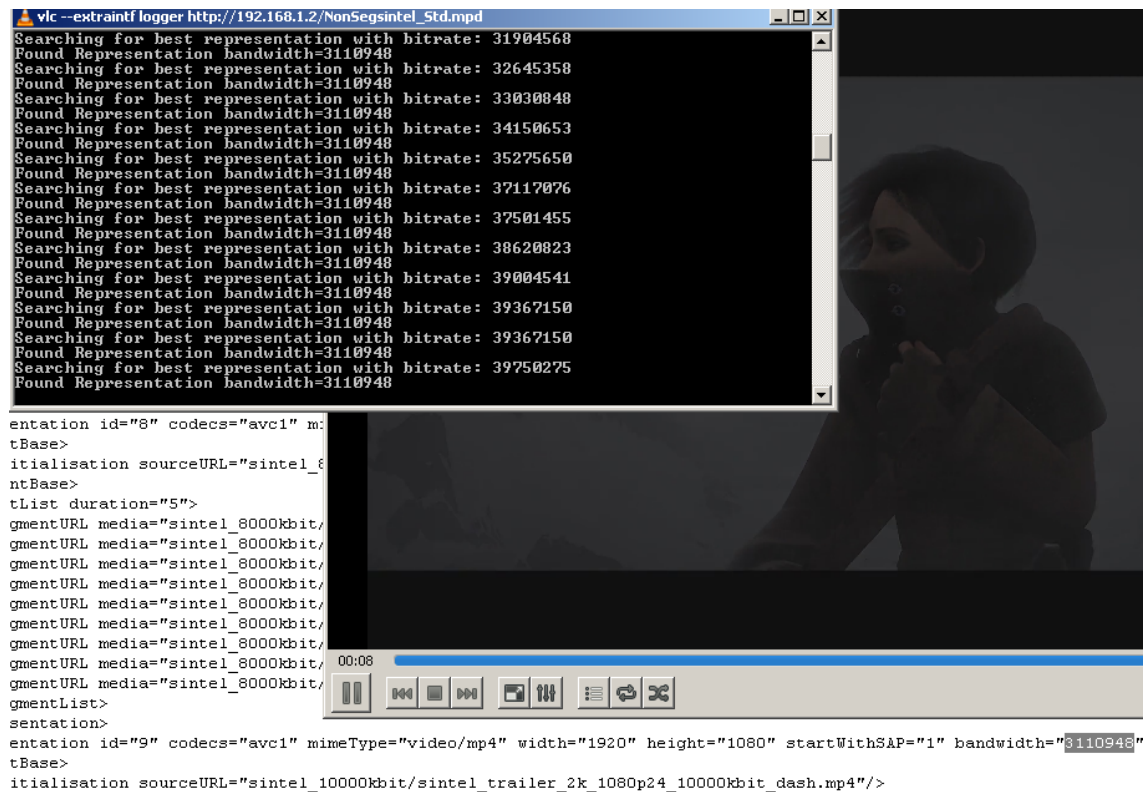


Fig. 4.5 Ample de banda seleccionat

A la figura 4.5 es pot veure en les línies que comencen per "Searching for..." com el bitrate observat continua pujant sense parar, amb uns valors superiors als 3.19 Mbps. Però llavors és quan baixem dràsticament l'ample de banda de 20 Mbps fins a 800Kbps, molt per sota del que actualment es necessitava per aquest stream però per sobre de la taxa mínima (200 kbps), i automàticament, el client DASH comença a baixar ràpidament com es pot observar a la figura 4.6.

```

vlc --extraintf logger http://192.168.1.2/sintel_Std.mpd
Found Representation bandwidth=3110948
Searching for best representation with bitrate: 3995264
Found Representation bandwidth=3110948
Searching for best representation with bitrate: 3738144
Found Representation bandwidth=3110948
Searching for best representation with bitrate: 3515095
Found Representation bandwidth=3110948
Searching for best representation with bitrate: 3334742
Found Representation bandwidth=3110948
Searching for best representation with bitrate: 2836732
Found Representation bandwidth=2315865
[00e91030] main input error: ES_OUT_SET<GROUP>PCR is called too late (pts_delay increased to 1000 ms)
[00e91030] main input error: ES_OUT_RESET_PCR called
Searching for best representation with bitrate: 2445502
Found Representation bandwidth=2315865
Searching for best representation with bitrate: 2357252
Found Representation bandwidth=2315865
Searching for best representation with bitrate: 2164364
Found Representation bandwidth=1311510
Searching for best representation with bitrate: 1950229
Found Representation bandwidth=1311510
Searching for best representation with bitrate: 1912116
Found Representation bandwidth=1311510

```

Fig. 4.6 Baixada d'ample de banda

A la figura 4.6 es pot veure com la baixada que s'està tenint en el bitrate, afecta directament als segment que s'escullen, passant de 10 Mbps (3110948 bps) en les primeres línies que comencen per "Found Representation...", als 3 Mbps (2315865 bps) directament, saltant-se molts altres segments que hi ha entre aquests dos, per tornar a saltar a un segment inferior de 1.5 Mbps (1311510 bps) ja que encara continua baixant. I encara que no es veu a la imatge, finalment arribarà a la qualitat de 600 Kbps, que és el màxim que pot donar aquest stream DASH amb l'ample de banda configurat amb Dummynet a 800 Kbps.

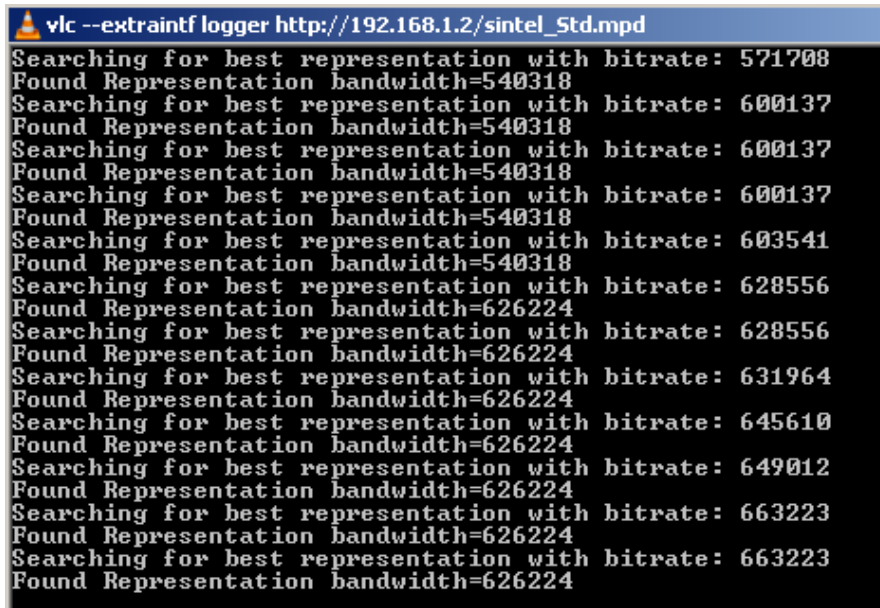
En quant a la relació entre bitrate observat, i el bitrate que s'està demanant per cada stream, la figura 4.7 permet veure clarament com el nostre bitrate observat és una mitja, mentre que el bitrate de l'stream s'escull mitjançant el pic màxim de bitrate. Aquesta captura s'ha fet utilitzant MediaInfo [31].

Format	: MPEG-4
Format profile	: Base Media
Codec ID	: isom
File size	: 19.5 MiB
Duration	: 52s 208ms
Overall bit rate	: 3 134 Kbps
Encoded date	: UTC 2012-06-13 22:12:12
Tagged date	: UTC 2012-06-13 22:12:12
Video	
ID	: 1
Format	: AVC
Format/Info	: Advanced Video Codec
Format profile	: Baseline@L5.0
Format settings, CABAC	: No
Format settings, ReFrames	: 5 frames
Format settings, GOP	: M=1, N=48
Codec ID	: avc1
Codec ID/Info	: Advanced Video Coding
Duration	: 52s 208ms
Bit rate	: 3 133 Kbps
Maximum bit rate	: 10.2 Mbps
Width	: 1 920 pixels
Height	: 1 080 pixels
Display aspect ratio	: 16:9
Frame rate mode	: Constant
Frame rate	: 24.000 fps

Fig. 4.7 Informació de l'stream de 10 Mbps amb MediaInfo

Finalment, es torna a pujar la configuració de la pipe a 1300 Kbps per arribar a 1 Mbps que és per sobre dels 600 Kbps observant com DASH torna a reaccionar i augmenta l'ample de banda. I, tal i com es pot veure a la figura 4.8, el bitrate observat comença a pujar lentament de 571708 bps fins a arribar a 663223 bps en la figura(i continuarà pujant encara més). A la figura doncs, es

torna a observar un canvi de selecció d'ample de banda de 540318 bps a 626224 bps.



```

vlc --extraintf logger http://192.168.1.2/sintel_Std.mpd
Searching for best representation with bitrate: 571708
Found Representation bandwidth=540318
Searching for best representation with bitrate: 600137
Found Representation bandwidth=540318
Searching for best representation with bitrate: 600137
Found Representation bandwidth=540318
Searching for best representation with bitrate: 600137
Found Representation bandwidth=540318
Searching for best representation with bitrate: 603541
Found Representation bandwidth=540318
Searching for best representation with bitrate: 628556
Found Representation bandwidth=626224
Searching for best representation with bitrate: 628556
Found Representation bandwidth=626224
Searching for best representation with bitrate: 631964
Found Representation bandwidth=626224
Searching for best representation with bitrate: 645610
Found Representation bandwidth=626224
Searching for best representation with bitrate: 649012
Found Representation bandwidth=626224
Searching for best representation with bitrate: 663223
Found Representation bandwidth=626224
Searching for best representation with bitrate: 663223
Found Representation bandwidth=626224

```

Fig. 4.8 Pujada d'ample de banda

Sense utilitzar Dummynet, la primera impressió sobre aquest número al final de la línia "Searching for..." va ser que només augmentava, augmentant l'ample de banda. Va ser a l'utilitzar Dummynet quan es va veure que aquest número era el bitrate observat. L'ample de banda comença pujant perquè és la finalitat de DASH, oferir un bon servei a l'usuari, i per tant sempre intentarà buscar un ample de banda millor dintre de les possibilitats de la connexió per oferir un stream millor. Però si en algun moment reduïm l'ample de banda per sota del necessari de la resolució que tenim configurada, l'adaptació respon correctament, baixant a un altre stream de la mateixa resolució però amb menys taxa. El problema per veure-ho és el buffer, que provoca que s'hagi d'esperar un temps en notar els efectes, ja que el buffer funciona precisament per solucionar petits problemes de xarxa. El buffer mínim per defecte en VLC és de 30 segons, encara que es pot modificar i en la majoria de les proves, s'ha utilitzat un buffer de 10 segons.

L'última prova que es va fer amb ample de banda va ser forçar un ample de banda de 150 Kbps quan el mínim bitrate per l'MPD és un segment de 200 Kbps. El resultat va ser coherent: tot i que VLC arrenca, i ho fa amb el segment de 200 Kbps, evidentment, no és capaç d'arribar al bitrate mínim i, encara que en cap moment es para, si que es veuen errors constants en la reproducció.

4.6.6. Provant amb el retard

Es va comprovar també la resposta del mòdul VLC amb DASH introduint retards (delay) amb Dummynet, amb l'objectiu d'observar el funcionament del buffer i el seu comportament real. I tot s'ha de dir, els resultats no han estat els esperats. Les proves de delay es van realitzar totes amb aquesta configuració:

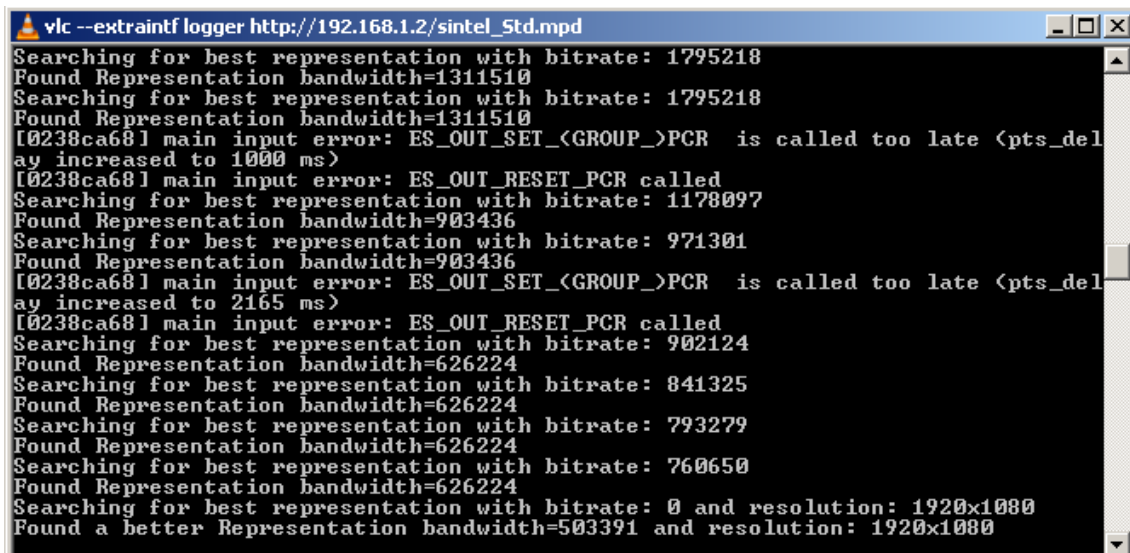
- Un ample de banda configurat de 6000 Kbps amb Dummynet.
- A la banda del servidor, una estructura definida per l'MPD mostrada a l'annex A.II.1, amb una mida de segment de 5 segons.
- El VLC configurat amb 10 segons de buffer mínim.

La primera prova que es va fer va ser introduir un petit delay des del principi: 50ms, abans inclús de demanar l'stream. Però aquesta prova ha resultat poc concloent. En introduir un delay des del principi, els paquets simplement tenen un retard inicial, però no hi ha cap alteració en el flux de dades, conseqüentment el vídeo simplement triga 50ms més en començar.

La segona prova, molt semblant i per poder veure la resposta a una alteració del flux per retard, primer es presenta el vídeo i, quan ja ha carregat un buffer mínim, s'introdueix el retard. Aquestes proves han resultat molt més interessants. Per començar, amb una prova similar a l'anterior, es va començar carregant el vídeo sense cap retard i uns segons més tard es provoca un petit retard de 50ms, però amb aquest retard tan petit VLC ni tan sols va fer mostra d'assabentar-se.

La cosa canvia quan passem directament a 4segons de delay. Aquí, passat el temps de buffer, VLC es penja sense poder recuperar-se del delay sofert. Tot i que els segments són de 5 segons i s'està introduït menys delay que la mida del segment, un delay tan gran deixa a VLC completament congelat. I possiblement, aquest tall sigui degut a algun timeout de TCP.

La següent prova va consistir en un delay de 400ms. Amb aquest delay passa una cosa molt interessant que ens mostra alguns detalls del funcionament del mòdul. La figura 4.9 mostra els resultats de l'experiment.



```

vlc --extraintf logger http://192.168.1.2/sintel_Std.mpd
Searching for best representation with bitrate: 1795218
Found Representation bandwidth=1311510
Searching for best representation with bitrate: 1795218
Found Representation bandwidth=1311510
[0238ca68] main input error: ES_OUT_SET_(GROUP_)PCR is called too late (pts_delay increased to 1000 ms)
[0238ca68] main input error: ES_OUT_RESET_PCR called
Searching for best representation with bitrate: 1178097
Found Representation bandwidth=903436
Searching for best representation with bitrate: 971301
Found Representation bandwidth=903436
[0238ca68] main input error: ES_OUT_SET_(GROUP_)PCR is called too late (pts_delay increased to 2165 ms)
[0238ca68] main input error: ES_OUT_RESET_PCR called
Searching for best representation with bitrate: 902124
Found Representation bandwidth=626224
Searching for best representation with bitrate: 841325
Found Representation bandwidth=626224
Searching for best representation with bitrate: 793279
Found Representation bandwidth=626224
Searching for best representation with bitrate: 760650
Found Representation bandwidth=626224
Searching for best representation with bitrate: 0 and resolution: 1920x1080
Found a better Representation bandwidth=503391 and resolution: 1920x1080

```

Fig. 4.9 Errors per delay en la transmissió

VLC es congela per uns segons i eventualment es recupera amb errors constants des d'aleshores i a un bitrate de 503391 bps. Com podem veure a la

figura, quan VLC observa que no arriba cap paquet, automàticament el bitrate observat baixa dràsticament fins a arribar a “observar” 0 bps i per tant el segment que s’escollirà a partir del delay començarà amb el segment d’ample de banda mínim (503391 bps). Aquesta part és completament lògica i esperada, però s’hauria de poder recuperar d’aquesta baixada de bitrate, amb un buffer de 10 segons, i no s’hauria de congelar o introduir els errors que es poden apreciar a la reproducció.

4.6.7. Provant amb pèrdues

Amb l’opció plr, que permet introduir una probabilitat de pèrdua de paquets amb Dummynet, es van fer un parell de proves. Començant amb una pèrdua del 0.05, o 5%, el sistema és massa inestable, inclús per 10 segons de buffer, aquest no arriba a recuperar-se de les contínues pèrdues de paquets. Es produeixen errors greus, congelació d’imatge i blocking constant.

La cosa canvia amb un 0.01 de pèrdua (1%). Aquest nivell, encara que altíssim per una bona comunicació, deixa al buffer omplir-se i durant la duració del vídeo només hi ha un parell de congelacions gens greus.

4.6.8. Provant amb jitter

El jitter és la variació del retard de transferència dels paquets, el que a la seva vegada produeix reordenacions i variacions del temps entre arribades. I tot i que hauria de ser més crític que el delay per la aleatorietat dels paquets, amb les proves que s’han fet (tot i que no són concloents), s’ha observat un resultat menys crític i és que amb 2 pipes configurades, una amb probabilitat de 90% i l’altre de 10%, se li aplica a la segona un delay de 500 ms, per comparar el resultat amb el resultat del delay.

El fet és que amb aquesta configuració, només hi ha errors constants, però no trobem la congelació que tenim amb el delay ni tan sols la baixada d’ample de banda.

4.6.9. Anàlisi del diàleg HTTP

També, com a fitxer adjunt, s’ha inclòs al projecte la captura de l’HTTP mitjançant Wireshark. DASH utilitza completament el protocol HTTP i es pot veure en tot moment les comandes GET de l’HTTP per demanar cada segment de vídeo cada cert temps com a la figura 4.10.

Filter: http			Expression...	Clear	Apply
No.	Time	Info			
8	0.005577	GET /sintel_std.mpd HTTP/1.1			
24	0.032422	GET /sintel_200kbit/sintel_trailer_2k_1080p24_200kbit_dash.mp4 HTTP/1.1			
35	0.051961	GET /sintel_200kbit/sintel1.m4s HTTP/1.1			
446	0.360605	GET /sintel_10000kbit/sintel2.m4s HTTP/1.1			
2303	7.188528	GET /sintel_10000kbit/sintel3.m4s HTTP/1.1			

Fig. 4.10 Comandes GET de HTTP

Una altra captura que es va fer, també va resultar molt interessant. Aquesta, es va realitzar durant les proves de Dummynet amb el video d'*Elephants Dream* [31] no fragmentat i segmentat en 4 segons.

Durant aquesta prova es va baixar de 2200 Kbps a 1100 Kbps després d'uns 5 segons de reproducció i després de 10 segons es va tornar a pujar a 2200 Kbps. Amb la captura de Wireshark es veu clarament aquesta diferència d'amples de banda.

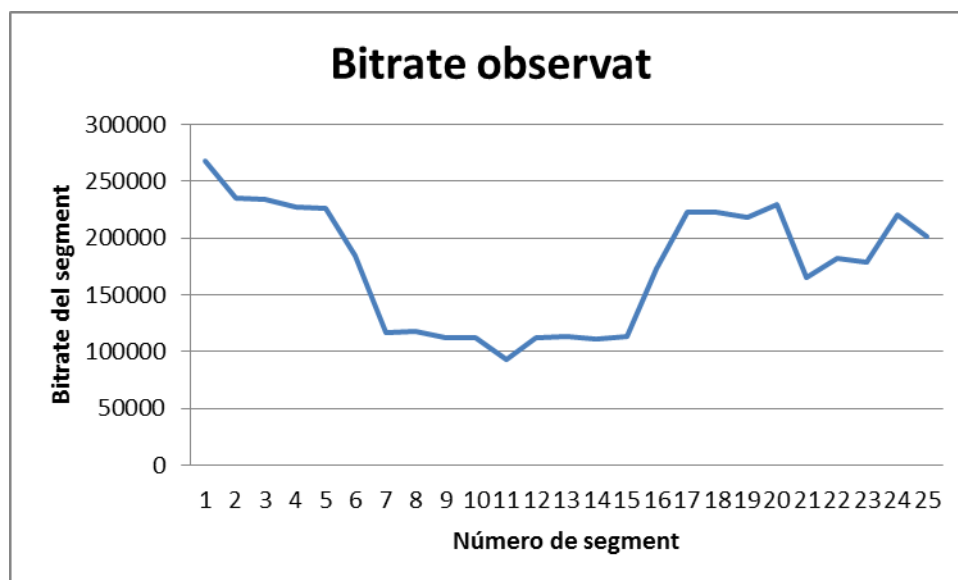


Fig. 4.11 Gràfic de l'ample de banda

Per fer aquest càlcul, s'ha agafat el rang del segment, que en tot moment ve indicat en la comanda GET de l'HTTP, amb el que s'obté el total de bytes del paquet, i es divideix pel temps que triga en arribar tot el rang de dades segmentats en paquets TCP. Aquest és el gràfic, en la figura 4.11, obtingut pel primer minut i mig del video durant el qual es van fer les proves. Com que la captura de Wireshark, a l'incloure tot i cada un dels paquets de la comunicació, és un llistat massa extens, inclús per l'annex, la captura del Wireshark s'adjunta en format digital.

4.6.10. DASH-JS amb Google Chrome

L'última setmana de març de 2012 va sortir una notícia [25] i una nova eina que és força interessant i promet molt pel futur de DASH. Es tracta de DASH-JS una eina feta en javascript per HTML5.

Aquesta eina s'integra amb el vídeo i permet reproduir MPD. Malauradament, després d'haver-ne fet proves, tot indica que DASH-JS encara no suporta el canvi de segments. Per arribar a aquesta conclusió, es van fer varies proves:

- Baixant el bitrate de 2700 Kbps a 1900 Kbps.
- Pujant el bitrate de 1900 Kbps un altre cop a 2700 Kbps.
- Baixant de 2700 Kbps a 500 Kbps.
- Baixant de 2700 Kbps a 500 Kbps i tornant a pujar a 1000 Kbps.
- Pujant de 800K bps a 2700 Kbps.

Malauradament, només els dos primers van arribar a funcionar, els tres últims deixaven Google Chrome congelat. Vam tornar a repetir la primera prova amb el Wireshark capturant. Tal i com es pot observar a la figura 4.12, que ha estat escurçada mostrant uns quants paquets de l'inici, del mig i del final, DASH-JS a seguit demanant el vídeo a 2400k, encara que s'hagués baixat l'ample de banda a 1700 Kbps.

Aquest comportament pot ser degut al buffer intern. DASH-JS ha d'intentar descarregar el mateix stream, i encara que baixem a 1700K bps, com que hi ha un buffer ple, la llibreria DASH-JS té prou temps per carregar el vídeo de 2400 Kbps, encara que no arribi a aquest ample de banda.

17	0.002247	GET / HTTP/1.1
21	0.003675	HTTP/1.1 304 Not Modified
22	0.059239	GET /favicon.ico HTTP/1.1
23	0.059800	HTTP/1.1 404 Not Found (text/html)
24	0.098959	GET /sintel_2400k.webm HTTP/1.1
26	0.099545	GET /sintel_2400k.webm HTTP/1.1
30	0.108757	HTTP/1.1 206 Partial Content (video/webm)
1358	4.042738	GET /sintel_2400k.webm HTTP/1.1
2288	6.780620	HTTP/1.1 206 Partial Content (video/webm)
2289	6.799399	GET /sintel_2400k.webm HTTP/1.1
2699	8.004366	HTTP/1.1 206 Partial Content (video/webm)
2701	8.022582	GET /sintel_2400k.webm HTTP/1.1
2799	8.301695	HTTP/1.1 206 Partial Content (video/webm)
2800	8.317156	GET /sintel_2400k.webm HTTP/1.1
11765	38.961029	HTTP/1.1 206 Partial Content (video/webm)
11767	39.310648	GET /sintel_2400k.webm HTTP/1.1
12237	40.933654	HTTP/1.1 206 Partial Content (video/webm)
12239	41.259540	GET /sintel_2400k.webm HTTP/1.1
12696	42.835441	HTTP/1.1 206 Partial Content (video/webm)
12698	43.159376	GET /sintel_2400k.webm HTTP/1.1
12896	43.838006	HTTP/1.1 206 Partial Content (video/webm)

Fig. 4.12 Exemple de captura de Wireshark

Per tant, a DASH-JS, encara li manca per poder veure un comportament adaptatiu real. De moment, tal i com mostren les dades, donat un ample de banda inicial, és capaç de reproduir el vídeo sencer amb el mateix segment, és a dir, només pot llegir l'MPD sense fer cap algoritme adaptatiu.

4.7. Altres proves

4.7.1. Contingut “Live”

També es va investigar sobre altres aspectes com streams en directe (*live*) per DASH, però fins ara, VLC ha estat explícit en només suportar contingut sota demanda (*on demand*). S’ha buscat informació per la xarxa sobre algun possible stream en directe amb DASH però no s’ha trobat res. Això demostra, una vegada més, que a les eines actuals de DASH li falten molts detalls encara.

4.7.2. Mozilla i LibDASH

L’última setmana del mes de maig de 2012 ha sortit la notícia [27] que també Mozilla desenvoluparà amb libDASH una ampliació per tal que Firefox accepti DASH. Aquest desenvolupament va sortir a la llum massa tard per realitzar les proves adients, i tot i que asseguraven que Mozilla només és capaç de reproduir MPDs tal i com fa DASH-JS, és una bona notícia que ambdós navegadors estiguin treballant d’aquesta manera per implementar DASH amb HTML5.

No només això, les *nightly builds* de Mozilla estan presents també per Android i l’estudiant ha pogut provar DASH sota Android després de configurar firefox tal i com s’explica a la secció d’eines 3.13.2 i carregant la pàgina de mozilla per fer proves amb DASH [32].

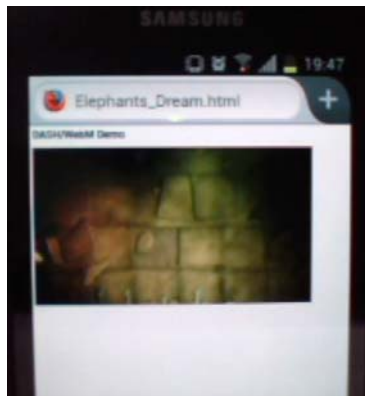


Fig. 4.13 Mozilla i DASH en Android

4.7.3. WebM

Tant Google Chrome com Firefox de Mozilla tenen una cosa en comú. L’ús del format WebM [33] per la reproducció de DASH. S’ha passat d’un format estandarditzat com el mp4 a WebM⁸, si això ha estat una bona o mala elecció, o si en un futur també s’utilitzaran altres formats és una cosa que només el temps aclarirà.

⁸ WebM és un format d’àudio i video desenvolupat sota una llicència BSD, i per tant, es tracta d’un format obert. Basat en Matroska[34] i propulsat per Google, aquest format ha estat àmpliament suportat per diferents navegadors.

CAPÍTOL 5. CONCLUSIONS I LÍNIES FUTURES

5.1. Conclusions

Aquest treball ha estat una molt bona oportunitat per l'autor per de posar-se al dia. Com a estudiant, és una oportunitat molt gran poder treballar en estàndards que encara estan oberts i subjectes a canvi, que encara no han sortit al "mercat". L'oportunitat d'aprenentatge, de veure com estan evolucionant les tecnologies i quins passos es segueix per aconseguir-ho. Aquest treball li ha donat la oportunitat a l'estudiant de conèixer no només una sinó dues tecnologies: DASH i MMT. Aquestes són tecnologies que pel seu enfoc i visió del món de les telecomunicacions, segur que donaran que parlar d'aquí ben poc.

Durant el treball s'han vist evolucionar moltes eines. Sense anar massa lluny, les evolucions del mòdul de VLC han estat una gran ajuda per l'estudiant, encara que s'han tret aspectes molt interessants, com el gràfic en consola. El reproductor MP4Client de GPAC també va patir diversos contratemps. Tots aquests contratemps van fer que es considerés de fer proves entre les diverses configuracions que podíem tenir entre el software de GPAC i el VLC. Actualment encara que també hi ha desavinences entre els dos reproductors, són molt més fluids i funcionen correctament al carregar els seus streams. També s'ha pogut observar durant el treball que l'adaptació de l'streaming mitjançant DASH és possible amb VLC. A més, apareixen cada poc temps nous clients per DASH com Google Chrome amb DASH-JS i Mozilla amb libDASH. Tanmateix, ara per ara els clients amb prou feines són capaços d'interpretar el MPD com per fer un streaming adaptatiu DASH.

Com a valoració d'aquest treball de fi de carrera, cal esmentar les coses positives i les coses negatives. Al principi, es pensava que MMT tindria una mica més de pes en aquest treball, però avui dia encara s'està acabant de definir moltes coses. Una de les altres coses, per part de l'alumne ha estat la documentació, tot i que al principi va intentar començar amb bon peu i seguir la documentació dia a dia en un bloc [28]. Finalment, com a coses positives, l'estudiant està molt satisfet amb tot allò que ha provat i fet en aquest projecte. Malgrat que la tecnologia més interessant, MMT, no ha pogut encaminar-se encara, l'estudiant també és conscient que encara resta per veure DASH amb unes eines 100% acabades, i encara més MMT, per tant sap que encara es podrien fer més proves o millorar software existent.

5.2. Impacte Mediambiental

DASH permet una utilització més òptima dels recursos. Encara que des d'un punt de vista dels servidors requereix més o menys el mateix espai per una estructura DASH amb fitxers no fragmentats que el que hi ha fins ara, des del punt de vista de l'usuari, la càrrega d'un stream òptim per la seva connexió

permet que es pugui balancejar millor una càrrega entre nodes, i també permetria menys consum d'energia en un terminal mòbil al poder escollir una resolució menor i, per tant, un consum de CPU menor. De totes maneres, la millora mediambiental seria marginal.

5.3. Línies futures

Actualment, les eines que hi ha en el mercat encara no estan desenvolupades, a totes aquestes eines encara li manquen funcionalitats i a algunes d'aquestes eines, funcionalitats molt bàsiques de DASH.

És clar que DASH encara li queda molt per ser un estàndard prou desenvolupat i utilitzat en el món. Recentment s'han llançat notícies en la pàgina web oficial [1] sobre nous projectes sobre DASH en javascript sobre Google Chrome i nativament sobre Mozilla Firefox de manera que es pugui reproduir sobre HTML5. Tanmateix aquest software encara li manca per estar acabat.

Tot i que s'han provat moltíssimes eines en aquest projectes i escenaris amb Dummynet, no ha donat temps per fer encara més proves i escenaris. Per exemple: com reacciona un terminal mòbil Android quan pateix els efectes propis del canal ràdio durant la recepció de l'stream DASH. També es podria estudiar la integració de DASH en un escenari CDN, desenvolupar una aplicació amb libDASH, o fins i tot provar amb diferents codificadors (no tan sols amb codificadors i contenidors H.26/MP4 o WebM).

Com a idea final, i donat el dinamisme en el camp de l'streaming adaptatiu, els treballs que actualment es realitzen sobre DASH, incloent aquest Treball de Final de Carrera, podrien ben bé no ser vàlids d'aquí uns mesos. Les eines que hi ha actualment han d'evolucionar molt encara, però aquest treball pretén ser una bona base perquè altres estudiants el puguin ampliar amb estudis posteriors.

BIBLIOGRAFIA

- [12] Pàgina web oficial de MPEG <http://www.mpeg.org/>
- [13] Pàgina web oficial de DVB, <http://www.dvb.org>
- [14] Introducció a MPEG-2 Stream, http://www.interactivetvweb.org/tutorials/dtv_intro/dtv_intro
- [15] Standard ISO/BMFF, http://standards.iso.org/ittf/PubliclyAvailableStandards/c051533_ISO_IEC_14496-12_2008.zip
- [16] Introducció a fitxers MPEG-4, <http://www.jiscdigitalmedia.ac.uk/audio/advice/aac-audio-and-the-mp4-media-format>
- [17] Standard DASH ISO/IEC 23009-1, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=57623
- [18] Viquipèdia sobre CDN, http://en.wikipedia.org/wiki/Content_delivery_network
- [19] Article sobre DASH streaming, http://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=79041&goback=%2Egde_2891647_member_113355128
- [20] MPD de Big Buck Bunny segmentat, http://www-itec.uni-klu.ac.at/ftp/datasets/mmsys12/BigBuckBunny/MPDs/BigBuckBunny_1s_isoffmain_DIS_23009_1_v_2_1c2_2011_08_30.mpd
- [21] MPD de Big Buck Bunny no segmentat, http://www-itec.uni-klu.ac.at/ftp/datasets/mmsys12/BigBuckBunny/bunny_1s/BigBuckBunnyNonSeg_1s_isoffmain_DIS_23009_1_v_2_1c2_2011_08_30.mpd
- [22] Pàgina web oficial de DASH, <http://www-itec.uni-klu.ac.at/dash/>
- [23] Pàgina web oficial de VideoLan, <http://www.videolan.org/>
- [24] Pàgina web oficial de GPAC, <http://gpac.wp.mines-telecom.fr>
- [25] DASHEncoder, http://www-itec.uni-klu.ac.at/dash/?page_id=282
- [26] Llista de correu sobre DASH, <http://lists.uni-klu.ac.at/mailman/listinfo/dash>
- [27] LibDASH, http://www-itec.uni-klu.ac.at/dash/?page_id=400
- [28] Pàgina web oficial d'Apache, <http://www.apache.org>
- [29] Pàgina web oficial de thttpd, <http://www.acme.com/software/thttpd/>
- [30] Pàgina web oficial de Dummynet, <http://info.iet.unipi.it/~luigi/dummynet/>

- [31] Demostració de canvi de sessió, <http://www.youtube.com/watch?v=f7UG6bfL3gw>
- [32] Pàgina web oficial d'x264, <http://www.videolan.org/developers/x264.html>
- [33] Pàgina web oficial de FrenzyBSD, <http://frenzy.org.ua/eng/>
- [34] Pàgina web oficial de WireShark, <http://www.wireshark.org/>
- [35] MPD Validator, http://www-itec.uni-klu.ac.at/dash/?page_id=605
- [36] DASH-JS, http://www-itec.uni-klu.ac.at/dash/?page_id=746
- [37] Pàgina web oficial d'HTML 5, <http://dev.w3.org/html5/html-author/>
- [38] DASH en Mozilla Firefox, <http://www-itec.uni-klu.ac.at/dash/?p=833>
- [39] Bloc creat per l'estudiant, <http://bitacoradpc.blogspot.com/>
- [40] Primer stream DASH provat, <http://www-itec.uni-klu.ac.at/dash/wp-content/dash/mtbike.mpd>
- [41] Fonts de vídeo, <http://media.xiph.org/>
- [42] Fonts de Elephants Dream per DASH, <ftp://ftp-itec.uni-klu.ac.at/pub/datasets/mmsys12/ElephantsDream/>
- [43] Exemple de pàgina en HTML5 per Mozilla, http://people.mozilla.com/~sworkman/Elephants_Dream.html
- [44] Especificació sobre el format WebM en DASH, <http://wiki.webmproject.org/adaptive-streaming/webm-dash-specification>
- [45] Pàgina web oficial de Matroska, <http://matroska.org/>
- [46] MMT Context & Objectives
- [47] MMT Working Draft: ISO/IEC CD 23008_1

[48] Altra documentació:

- [49] Fitxer públic XSD, http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-DASH_schema_files/DASH-MPD.xsd
- [50] DASH en la Viquipèdia, http://en.wikipedia.org/wiki/Dynamic_Adaptive_Streaming_over_HTTP
- [51] Blog de Christian Timmerer, <http://multimediacommunication.blogspot.com.es/2010/05/http-streaming-of-mpeg-media.html>

- [52] Blog de Christian Timmerer II, <http://multimediacommunication.blogspot.com.es/2010/08/mmsys11-special-session-on-modern-media.html>
- [53] Document de presentació de DASH, <http://www-itec.uni-klu.ac.at/bib/files/STDay-2010-v1.1.pdf>
- [54] Document de disseny de l'estàndard, http://www.w3.org/2010/11/web-and-tv/papers/webtv2_submission_64.pdf
- [55] Primera Presentació oficial, <http://www.slideshare.net/christian.timmerer/dynamic-adaptive-streaming-over-http-dash>
- [56] Presentació de disseny de l'estàndard, http://web.cs.wpi.edu/~claypool/mmsys-2011/Day2-1_3GPPDynamic.pdf
- [57] Presentació de bases i prototipus, <http://www.slideshare.net/christian.timmerer/http-streaming-of-mpeg-media>
- [58] Presentació del prototipus de GPAC, http://web.cs.wpi.edu/~claypool/mmsys-2011/Day3-6_UsagesofDash.pdf
- [59] Presentació resum de DASH, http://web.cs.wpi.edu/~claypool/mmsys-2011/Day3-7_Testbed.pdf
- [60] Presentació DRM sobre DASH, http://web.cs.wpi.edu/~claypool/mmsys-2011/Day3-8_DRM.pdf
- [61] Prototipus en VLC, <http://www.youtube.com/watch?v=Yt1F0ULsA1w>
- [62] Demostració de VLC, <http://www.youtube.com/watch?v=TLBtzMqdKfk&feature=related>
- [63] Manual THHTTPD, <http://pbraun.nethence.com/doc/www/thhttpd.html>
- [64] Tutorial actual de Dummynet, <http://info.iet.unipi.it/~luigi/papers/20100304-ccr.pdf>
- [65] Tutorial original de Dummynet, http://info.iet.unipi.it/~luigi/ip_dummynet/original.html
- [66] Exemple de Dummynet I, <http://lists.freebsd.org/pipermail/freebsd-ipfw/2004-March/001007.html>
- [67] Exemple de Dummynet II, <http://lists.freebsd.org/pipermail/freebsd-ipfw/2004-March/000976.html>
- [68] Document d'una implementació Dummynet, http://enr.case.edu/liberatore_vincenzo/NetBots/wan_emu_howto.pdf
- [69] Documentació X264, http://mewiki.project357.com/wiki/X264_Settings
- [70] Documentació alternativa de X264, <http://www.mplayerhq.hu/DOCS/HTML/en/menc-feat-x264.html>

[71] Pàgina web oficial de FileZilla, <http://filezilla-project.org/>

GLOSSARI

CDN	Content Delivery Network
CPU	Central Processing Unit
DASH	Dynamic Adaptive Streaming over HTTP
DVB	Digital Video Broadcasting
HTTP	Hypertext Transfer Protocol
ISO	International Organization for Standardization
ISOBMFF	ISO Base Media File Format
MMT	MPEG Media Transport
MPD	Media Presentation Descriptor
MPEG	Moving Picture Experts Group
NAL	Network Abstraction Layer
PS	Program Stream
QoE	Quality of Experience
QoS	Quality of Service
RAP	Random Access Point
TCP	Transmission Control Protocol
TS	Transport Stream
XML	Extensible Markup Language
XSD	XML Schema Definition

ANNEXOS

A.I. MPEG MEDIA TRANSPORT (MMT)

A.I.1. Informació general

Els estàndards de codificació com MPEG-2 o MPEG-4, van ser desenvolupats sense pensar en la interaccions entre xarxes, els diferents dispositius i les seves capacitats, ja que llavors no existia la diversitat de xarxes d'accés i de terminals que hi ha actualment. Avui dia, la proliferació i l'ús de tecnologies mòbils ens ha portat a trobar els estàndards massa restrictius. Es vol una tecnologia completament transparent per l'usuari, i accions com veure al dispositiu mòbil al mateix *stream* que s'està veient per Televisió –atenció que no es parla del mateix contingut, sinó del mateix *stream*-, per exemple, ajudaria tant a les empreses com a l'usuari a trobar la tecnologia més transparent i potent. Però aquesta particularitat és només una de les que s'han volgut potenciar amb MMT (MPEG Media Transport), el nou estàndard en desenvolupament.

Així doncs, partint de la base del TS, amb un bon grapat de codificadors, tant d'àudio com de vídeo i amb la previsió que n'hi hagin molts més. MMT es situa en un àmbit d'actuació bastant gran. Intenta superar el TS en quant a capes afectades però sense entrar a definir res de la capa de codificació. MMT pretén ser un protocol per la transmissió de tot tipus de multimèdia, i com a tal, hauria de poder controlar tots els paràmetres de transmissió.

A.I.2. Característiques

Amb MMT es pretén definir un nou protocol que permeti varies funcionalitats, entre les quals les més importants són:

- Una distribució de qualitat independent del tipus de xarxa a la que estigui connectat el dispositiu.
- Permetre informació creuada entre capes de manera que es pugui accedir a informació clau per adaptar la transmissió i aconseguir una QoS òptima.
- Permetre l'ús de diferents xarxes simultànies pel consum d'un servei com podria ser, per exemple, TV interactiva emetent vídeo i àudio per un canal de difusió unidireccional com ara la TDT i les dades per un canal bidireccional com ara la xarxa IP.
- Permetre l'ús de serveis amb poca latència com ara jocs online.
- Millora de la distribució, senyalització i també l'ús d'eines de protecció de drets.
- Permetre la redistribució dels continguts per optimitzar l'*streaming*.

- Millorar la distribució un-a-molts.
- Millorar la protecció contra errors.

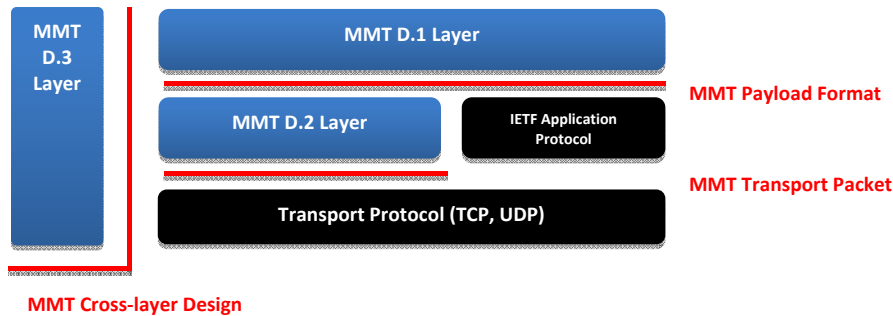


Fig. A.1 Àmbit d'actuació del MMT. Extret de [30]

Això s'aconsegueix mitjançant un conjunt de capes definides per comunicar-se entre elles inclús amb les capes més inferiors de la comunicació. D'aquesta manera, controlar els errors que hi ha a l'stream és més senzill ja que hi ha més eines per saber quin error real està sofrint la comunicació. L'ajust de varies capes permet a cada una especialitzar-se i poder controlar millor el que pot i donar informació a aquelles altres capes que necessitin d'aquesta informació.

Com a es pot observar a la figura A.1, aquest protocol pretén estar al mateix nivell que RTP inclús la capa D.3 arribaria al nivell TCP/UDP. La raó d'aquest fet és, com s'ha dit abans, per controlar variables que poden ser crucials per el control i la millora de la qualitat de servei (QoS).

A.I.3. Evolució de l'estàndard

Des que es va començar aquest projecte, l'estàndard MMT encara s'està modificant. És per aquests motius que no s'ha pogut dedicar-hi massa temps a aquest estàndard. L'exemple idoni d'aquesta variació en l'estàndard és l'àmbit d'actuació del MMT. Com es veu a la figura A.2, l'àmbit de les capes D2 i D1 eren per sobre de la capa de Transport TCP, mentre que abans MMT pretenia inclús substituir el nivell TCP amb la capa D2:

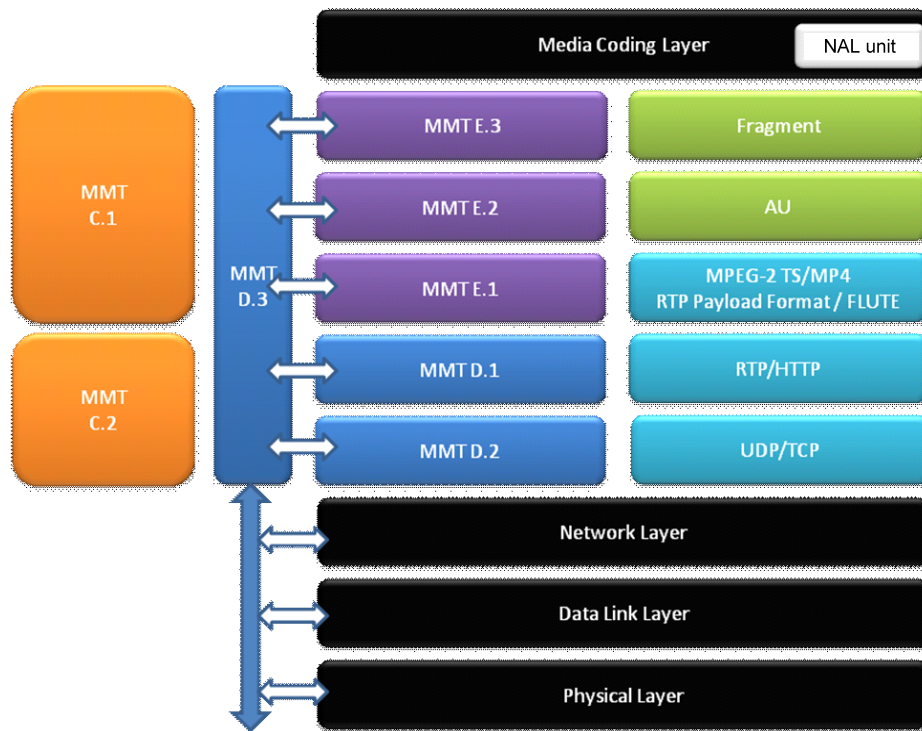


Fig. A.2 Àrees funcionals de MMT. Extret de [35]

A.I.4. Entrant en l'estàndard

A MMT es defineixen 3 capes d'actuació: encapsulament (E), transmissió (D) i control (C), cadascuna amb funcions específiques, inclús amb diferents nivells d'actuació i amb interacció entre elles. D'aquesta manera a la capa d'encapsulament tenim fins a 3 nivells diferents. I segons cada nivell tenim una unitat d'encapsulament diferent.

- En el primer nivell E.1 tenim els M-Unit, que conté els paquets generats per la capa superior, externa a MMT. Cada paquet, a partir d'ara MFU (Media Fragment Unit), s'encapsula en una capçalera que l'identifica dels altres MFU i dona prioritats i l'ordre de descodificació de cada MFU.
- En el segon, E.2, amb els MMT-Asset, es troben un o més M-Units, cadascun identificats individualment i també es troba informació d'accés aleatori: el Random Access Point (RAP).
- Finalment el tercer nivell amb E.3 tenim els MMT- Paquets. En aquest es troba un conjunt de MMT-Assets, que no són més que els components del multimèdia (video, àudio, informació del programa). Però també conté molta més informació, com la informació de configuració de cada MMT-Asset, on trobem característiques d'ordenació i d'identificació, també es troba una llista de tots els Assets disponibles, i finalment la configuració espacial i temporal de cada Asset (podria ser que un programa en Java compartís pantalla amb la Televisió, per exemple).

Per la part de la capa de Distribució, es poden identificar diferents funcionalitats, encara que no estan tan identificades per nivells com la capa d'encapsulament. Això és degut a que aquesta capa pretén que hi hagi comunicació entre totes les altres, entre la capa d'encapsulament, el nivell de transport i la capa de control per tal que es pugui modificar paràmetres de la sessió i així millorar el QoS.

- Funcions bàsiques com diferent tractament dels paquets de la capa E (aquí estariem parlant del nivell D.1), relacions temporals entre els paquets de la capa E per poder ajustar els paquets que s'envien per diferents xarxes, estimació del jitter de la xarxa.
- Funcions de control d'errors amb els típics FEC i possibles retransmissions i control de flux amb classificació de paquets segons bitrate, tipus i prioritats per evitar saturació de paquets(aquest seria el nivell D.2).
- Funcions de control de QoS. I com que el tema de QoS es massa ampli, aquesta part hauria de comunicar-se eficaçment amb tots els nivells de la transmissió (finalment el nivell D.3). Per aquesta raó s'implementa el Common Layer Interface, un conjunt d'interfícies de comunicació per cada capa.

En definitiva, estem parlant de la capa que ofereix més interacció entre les diferents capes. Això és, principalment, el major canvi que es pretenia fer amb MMT.

Però encara falta la capa de Control, una capa que ens permetrà configurar el flux MMT de diferents maneres i controlar aquest flux mitjançant dos nivells clarament anivellats amb les altres dues capes.

- Amb la capa C.1 tenim informació sobre el consum dels MMT-Assets. Senyalització de començament i finalització del consum del flux així com també control sobre diferent activitats sobre el flux com rebobinar, pausa, avançar, etc.
- La capa C.2 anivellada amb la capa D, pretén oferir una senyalització entre punts per tal d'oferir suport en aquestes funcions de la capa D. Per aquest motiu s'han creat una sèrie de missatges tots inclosos en la capa C.2 per tal de fer possible el que es pretén amb la capa D. Missatges per l'establiment de la sessió i el control d'aquesta a nivell de control d'errors, control de flux, etc. Missatges per la coordinació entre diferents tipus de xarxes (radiofreqüència, Ethernet, Wi-Fi, etc).

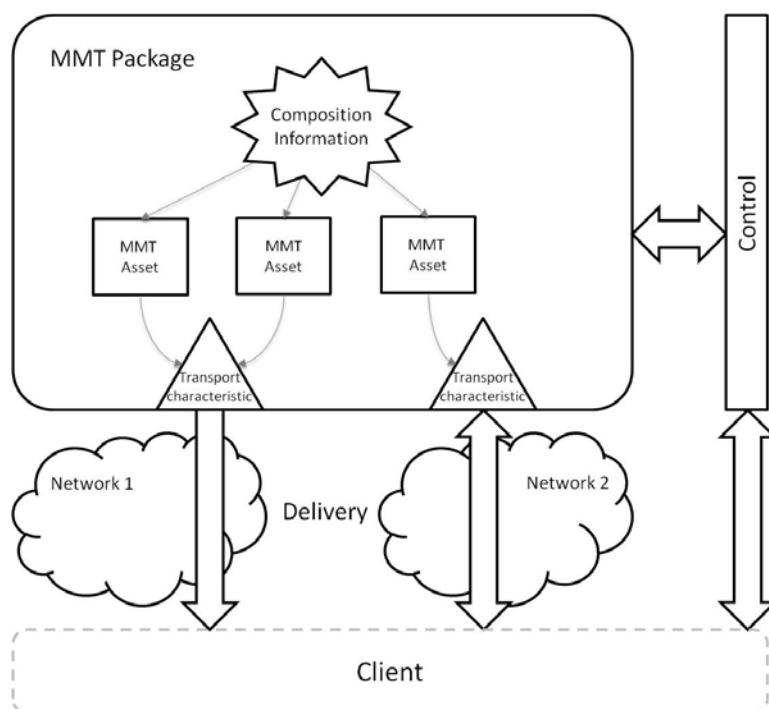


Fig. A.3 Coordinació entre diferents xarxes i nivell de la capa de control. Extret de [36]

A.II. Media Presentation Descriptor

A.II.1. Exemple de l'estudiant

L'MPD presentat en aquest apartat, està descrit a l'apartat 4.6.4.

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:DASH:schema:MPD:2011"
  xsi:schemaLocation="urn:mpeg:DASH:schema:MPD:2011"
  profiles="urn:mpeg:dash:profile:isoff-main:2011"
  type="static"
  mediaPresentationDuration="PT0M52S"
  minBufferTime="PT5.0S">
  <BaseURL>http://192.168.1.2/</BaseURL>
  <Period start="PT0S">
    <AdaptationSet bitstreamSwitching="true">
      <Representation id="0" codecs="avc1" mimeType="video/mp4" width="1920" height="1080" startWithSAP="1"
        bandwidth="503391" minBufferTime="2000">
        <SegmentBase>
          <Initialisation sourceURL="sintel_200kbit/sintel_trailer_2k_1080p24_200kbit_dash.mp4"/>
        </SegmentBase>
        <SegmentList duration="5">
          <SegmentURL media="sintel_200kbit/sintel1.m4s"/>
          <SegmentURL media="sintel_200kbit/sintel2.m4s"/>
          <SegmentURL media="sintel_200kbit/sintel3.m4s"/>
          <SegmentURL media="sintel_200kbit/sintel4.m4s"/>
          <SegmentURL media="sintel_200kbit/sintel5.m4s"/>
          <SegmentURL media="sintel_200kbit/sintel6.m4s"/>
          <SegmentURL media="sintel_200kbit/sintel7.m4s"/>
          <SegmentURL media="sintel_200kbit/sintel8.m4s"/>
          <SegmentURL media="sintel_200kbit/sintel9.m4s"/>
        </SegmentList>
      </Representation>
      <Representation id="1" codecs="avc1" mimeType="video/mp4" width="1920" height="1080" startWithSAP="1"
        bandwidth="540318" minBufferTime="2000">
        <SegmentBase>
          <Initialisation sourceURL="sintel_400kbit/sintel_trailer_2k_1080p24_400kbit_dash.mp4"/>
        </SegmentBase>
        <SegmentList duration="5">
          <SegmentURL media="sintel_400kbit/sintel1.m4s"/>
          <SegmentURL media="sintel_400kbit/sintel2.m4s"/>
          <SegmentURL media="sintel_400kbit/sintel3.m4s"/>
          <SegmentURL media="sintel_400kbit/sintel4.m4s"/>
          <SegmentURL media="sintel_400kbit/sintel5.m4s"/>
          <SegmentURL media="sintel_400kbit/sintel6.m4s"/>
          <SegmentURL media="sintel_400kbit/sintel7.m4s"/>
          <SegmentURL media="sintel_400kbit/sintel8.m4s"/>
          <SegmentURL media="sintel_400kbit/sintel9.m4s"/>
        </SegmentList>
      </Representation>
      <Representation id="2" codecs="avc1" mimeType="video/mp4" width="1920" height="1080" startWithSAP="1"
        bandwidth="626224" minBufferTime="2000">
        <SegmentBase>
          <Initialisation sourceURL="sintel_600kbit/sintel_trailer_2k_1080p24_600kbit_dash.mp4"/>
        </SegmentBase>
        <SegmentList duration="5">
          <SegmentURL media="sintel_600kbit/sintel1.m4s"/>
          <SegmentURL media="sintel_600kbit/sintel2.m4s"/>
          <SegmentURL media="sintel_600kbit/sintel3.m4s"/>
          <SegmentURL media="sintel_600kbit/sintel4.m4s"/>
          <SegmentURL media="sintel_600kbit/sintel5.m4s"/>
          <SegmentURL media="sintel_600kbit/sintel6.m4s"/>
          <SegmentURL media="sintel_600kbit/sintel7.m4s"/>
          <SegmentURL media="sintel_600kbit/sintel8.m4s"/>
          <SegmentURL media="sintel_600kbit/sintel9.m4s"/>
        </SegmentList>
      </Representation>
      <Representation id="3" codecs="avc1" mimeType="video/mp4" width="1920" height="1080" startWithSAP="1"
        bandwidth="903436" minBufferTime="2000">
        <SegmentBase>
```



```

    <Initialisation sourceURL="sintel_1000kbit/sintel_trailer_2k_1080p24_1000kbit_dash.mp4"/>
  </SegmentBase>
  <SegmentList duration="5">
    <SegmentURL media="sintel_1000kbit/sintel1.m4s"/>
    <SegmentURL media="sintel_1000kbit/sintel2.m4s"/>
    <SegmentURL media="sintel_1000kbit/sintel3.m4s"/>
    <SegmentURL media="sintel_1000kbit/sintel4.m4s"/>
    <SegmentURL media="sintel_1000kbit/sintel5.m4s"/>
    <SegmentURL media="sintel_1000kbit/sintel6.m4s"/>
    <SegmentURL media="sintel_1000kbit/sintel7.m4s"/>
    <SegmentURL media="sintel_1000kbit/sintel8.m4s"/>
    <SegmentURL media="sintel_1000kbit/sintel9.m4s"/>
  </SegmentList>
</Representation>
<Representation id="4" codecs="avc1" mimeType="video/mp4" width="1920" height="1080" startWithSAP="1"
bandwidth="1311510" minBufferTime="2000">
  <SegmentBase>
    <Initialisation sourceURL="sintel_1500kbit/sintel_trailer_2k_1080p24_1500kbit_dash.mp4"/>
  </SegmentBase>
  <SegmentList duration="5">
    <SegmentURL media="sintel_1500kbit/sintel1.m4s"/> <SegmentURL media="sintel_1500kbit/sintel2.m4s"/>
    <SegmentURL media="sintel_1500kbit/sintel3.m4s"/>
    <SegmentURL media="sintel_1500kbit/sintel4.m4s"/>
    <SegmentURL media="sintel_1500kbit/sintel5.m4s"/>
    <SegmentURL media="sintel_1500kbit/sintel6.m4s"/>
    <SegmentURL media="sintel_1500kbit/sintel7.m4s"/>
    <SegmentURL media="sintel_1500kbit/sintel8.m4s"/>
    <SegmentURL media="sintel_1500kbit/sintel9.m4s"/>
  </SegmentList>
</Representation>
<Representation id="5" codecs="avc1" mimeType="video/mp4" width="1920" height="1080" startWithSAP="1"
bandwidth="2315865" minBufferTime="2000">
  <SegmentBase>
    <Initialisation sourceURL="sintel_3000kbit/sintel_trailer_2k_1080p24_3000kbit_dash.mp4"/>
  </SegmentBase>
  <SegmentList duration="5">
    <SegmentURL media="sintel_3000kbit/sintel1.m4s"/>
    <SegmentURL media="sintel_3000kbit/sintel2.m4s"/>
    <SegmentURL media="sintel_3000kbit/sintel3.m4s"/>
    <SegmentURL media="sintel_3000kbit/sintel4.m4s"/>
    <SegmentURL media="sintel_3000kbit/sintel5.m4s"/>
    <SegmentURL media="sintel_3000kbit/sintel6.m4s"/>
    <SegmentURL media="sintel_3000kbit/sintel7.m4s"/>
    <SegmentURL media="sintel_3000kbit/sintel8.m4s"/>
    <SegmentURL media="sintel_3000kbit/sintel9.m4s"/>
  </SegmentList>
</Representation>
<Representation id="6" codecs="avc1" mimeType="video/mp4" width="1920" height="1080" startWithSAP="1"
bandwidth="2846768" minBufferTime="2000">
  <SegmentBase>
    <Initialisation sourceURL="sintel_4500kbit/sintel_trailer_2k_1080p24_4500kbit_dash.mp4"/>
  </SegmentBase>
  <SegmentList duration="5">
    <SegmentURL media="sintel_4500kbit/sintel1.m4s"/>
    <SegmentURL media="sintel_4500kbit/sintel2.m4s"/>
    <SegmentURL media="sintel_4500kbit/sintel3.m4s"/>
    <SegmentURL media="sintel_4500kbit/sintel4.m4s"/>
    <SegmentURL media="sintel_4500kbit/sintel5.m4s"/>
    <SegmentURL media="sintel_4500kbit/sintel6.m4s"/>
    <SegmentURL media="sintel_4500kbit/sintel7.m4s"/>
    <SegmentURL media="sintel_4500kbit/sintel8.m4s"/>
    <SegmentURL media="sintel_4500kbit/sintel9.m4s"/>
  </SegmentList>
</Representation>
<Representation id="7" codecs="avc1" mimeType="video/mp4" width="1920" height="1080" startWithSAP="1"
bandwidth="3005495" minBufferTime="2000">
  <SegmentBase>
    <Initialisation sourceURL="sintel_6000kbit/sintel_trailer_2k_1080p24_6000kbit_dash.mp4"/>
  </SegmentBase>
  <SegmentList duration="5">
    <SegmentURL media="sintel_6000kbit/sintel1.m4s"/>
    <SegmentURL media="sintel_6000kbit/sintel2.m4s"/>
    <SegmentURL media="sintel_6000kbit/sintel3.m4s"/>
    <SegmentURL media="sintel_6000kbit/sintel4.m4s"/>
    <SegmentURL media="sintel_6000kbit/sintel5.m4s"/>

```

```
<SegmentURL media="sintel_6000kbit/sintel6.m4s"/>
<SegmentURL media="sintel_6000kbit/sintel7.m4s"/>
<SegmentURL media="sintel_6000kbit/sintel8.m4s"/>
<SegmentURL media="sintel_6000kbit/sintel9.m4s"/>
</SegmentList>
</Representation>
<Representation id="8" codecs="avc1" mimeType="video/mp4" width="1920" height="1080" startWithSAP="1"
bandwidth="3108752" minBufferTime="2000">
<SegmentBase>
  <Initialisation sourceURL="sintel_8000kbit/sintel_trailer_2k_1080p24_8000kbit_dash.mp4"/>
</SegmentBase>
<SegmentList duration="5">
  <SegmentURL media="sintel_8000kbit/sintel1.m4s"/>
  <SegmentURL media="sintel_8000kbit/sintel2.m4s"/>
  <SegmentURL media="sintel_8000kbit/sintel3.m4s"/>
  <SegmentURL media="sintel_8000kbit/sintel4.m4s"/>
  <SegmentURL media="sintel_8000kbit/sintel5.m4s"/>
  <SegmentURL media="sintel_8000kbit/sintel6.m4s"/>
  <SegmentURL media="sintel_8000kbit/sintel7.m4s"/>
  <SegmentURL media="sintel_8000kbit/sintel8.m4s"/>
  <SegmentURL media="sintel_8000kbit/sintel9.m4s"/>
</SegmentList>
</Representation>
<Representation id="9" codecs="avc1" mimeType="video/mp4" width="1920" height="1080" startWithSAP="1"
bandwidth="3110948" minBufferTime="2000">
<SegmentBase>
  <Initialisation sourceURL="sintel_10000kbit/sintel_trailer_2k_1080p24_10000kbit_dash.mp4"/>
</SegmentBase>
<SegmentList duration="5">
  <SegmentURL media="sintel_10000kbit/sintel1.m4s"/>
  <SegmentURL media="sintel_10000kbit/sintel2.m4s"/>
  <SegmentURL media="sintel_10000kbit/sintel3.m4s"/>
  <SegmentURL media="sintel_10000kbit/sintel4.m4s"/>
  <SegmentURL media="sintel_10000kbit/sintel5.m4s"/>
  <SegmentURL media="sintel_10000kbit/sintel6.m4s"/>
  <SegmentURL media="sintel_10000kbit/sintel7.m4s"/>
  <SegmentURL media="sintel_10000kbit/sintel8.m4s"/>
  <SegmentURL media="sintel_10000kbit/sintel9.m4s"/>
</SegmentList>
</Representation>
  </AdaptationSet>
</Period>
</MPD>
```

A.III. DASHEncoder

A.III.1. El fitxer de configuració

DASHEncoder utilitza un fitxer de configuració bastant extens. A continuació presentem un exemple i el comentem. .

```
#=====
#   Config File for DASH Encoder
#=====

#=====
#       General Options
#=====

dest-directory : /home/dani/Escritorio/ExDASHEncoder/Ex9/

video-encoder : x264
audio-encoder : ffmpegAAC
multiplexer : mp4box

#store-psnr
sql-host : localhost
sql-user : root
sql-pw :
sql-database : dash

add-non-segmented

#use-ffmpeg-pipe
ffmpeg-opt : -f rawvideo
input-res : 1920x1080
#input-res : 384x208
#input-res : 352x288
#input-res : 854x480

#=====
#       x264 Options
#=====

bitrate : 1000

#bitrate :
50@352x288|100@352x288|150@352x288|200@352x288|250@352x288|300@352x288|400@854x480|500@85
4x480|600@854x480|700@854x480|800@854x480|1000@1920x1080|2000@1920x1080|3000@1920x1080|4000
@1920x1080|6000@1920x1080
#bitrate : 200|400|600|1000
statistics : stat.temp
gop : 48
scenecut : 0
profile : baseline
preset : slow
input : /home/dani/Escritorio/snow_mnt_1080p.y4m

passes : 1
const-filesize : 0

#Additional Options for Encoding

#pass1 : --frames 500 --fps 29.970029
pass1 : --verbose --fps 24
pass2 : --verbose --psnr
```

```

=====
#           FFMPEG AAC Options
=====

# [channels, samplerate, bitrate]
#audio-quality : 2,44100,48|2,44000,128
#audio-quality : 2,44100,48
audio-quality :

audio-input : /home/stefan/sintel_trailer-audio.flac
audio-codec : libfaac

=====
#           MP4Box Options
=====

#AV Muxing
#mux-combi : 300@48|600@128
mux-combi :

fragment-size : 2000
segment-size : 2000
rap-aligned
segment-name : snow
folder-prefix : snow

=====
#           MPD Options
=====
mpd-name : Snow.mpd
url-root : http://192.162.1.143/MPDs/

set-base-url
# --> With set-base-url:
#<SegmentList duration="2">
#  <SegmentURL media="snow_3000kbit/snow1.m4s"/>
#</SegmentList>
# --> Without set-base-url:
#<SegmentList duration="2">
#  <SegmentURL media="http://192.162.1.143/MPDs/snow_800kbit/snow1.m4s"/>
#  <SegmentURL media="http://192.162.1.143/MPDs/snow_800kbit/snow2.m4s"/>
#  ...

mpdActStandardPostfix : Std.mpd
duration : 0H0M21.54S
transform-mpd
minBufferTime : 2.0S
segDuration : 2
=====
#           Other Options
=====

```

Una de les coses interessants en aquest fitxer de configuració és veure com s'especifiquen les parts involucrades en crear l'MPD:

```

video-encoder : x264
audio-encoder : ffmpegAAC
multiplexer   : mp4box

```

Això dona a entendre que si es volgués utilitzar un altre codificador, DASHEncoder hauria de ser compatible amb altres opcions, tant video/audio codificadors, com el multiplexer tot i així, és una prova que no s'ha realitzat.

La part, però, més interessant està en la part del codificador x264, el paràmetre que més utilitat té d'aquesta part i possiblement del fitxer de configuració sencer és el bitrate. Es pot observar a la figura com hi ha tres línies, però dues d'elles estan comentades (hi ha un símbol # al començament de la línia). Aquests són exemples del que es pot fer amb el paràmetre bitrate de DASHEncoder per tal que generi streams DASH a les resolucions i bitrates desitjades automàticament:

- Es pot definir un sol bitrate a la resolució original (que prèviament s'ha assignat amb el paràmetre input-res):
 - bitrate : 1000
- Definir diferents bitrates amb la resolució original:
 - #bitrate : 200|400|600|1000
- Ó bé definir diferents bitrates a diferents resolucions:
 - #bitrate : 50@352x288|500@854x480|4000@1920x1080

Amb això DASHEncoder crearà un stream DASH per cada bitrate especificat.